

McClellan AFB
Contract #: F04606-85-C0733
CDRL Item: E002
24 May 90

**SOFTWARE DELIVERY DOCUMENTATION
FOR THE
AN/FMQ-13(V)
DIGITAL WIND SENSOR
APPLICATION PROGRAM**

CPIN: 83M-FMQ13-F002-00A
ENGR SPEC #: MMA 81-025B
REVISION: INITIAL RELEASE

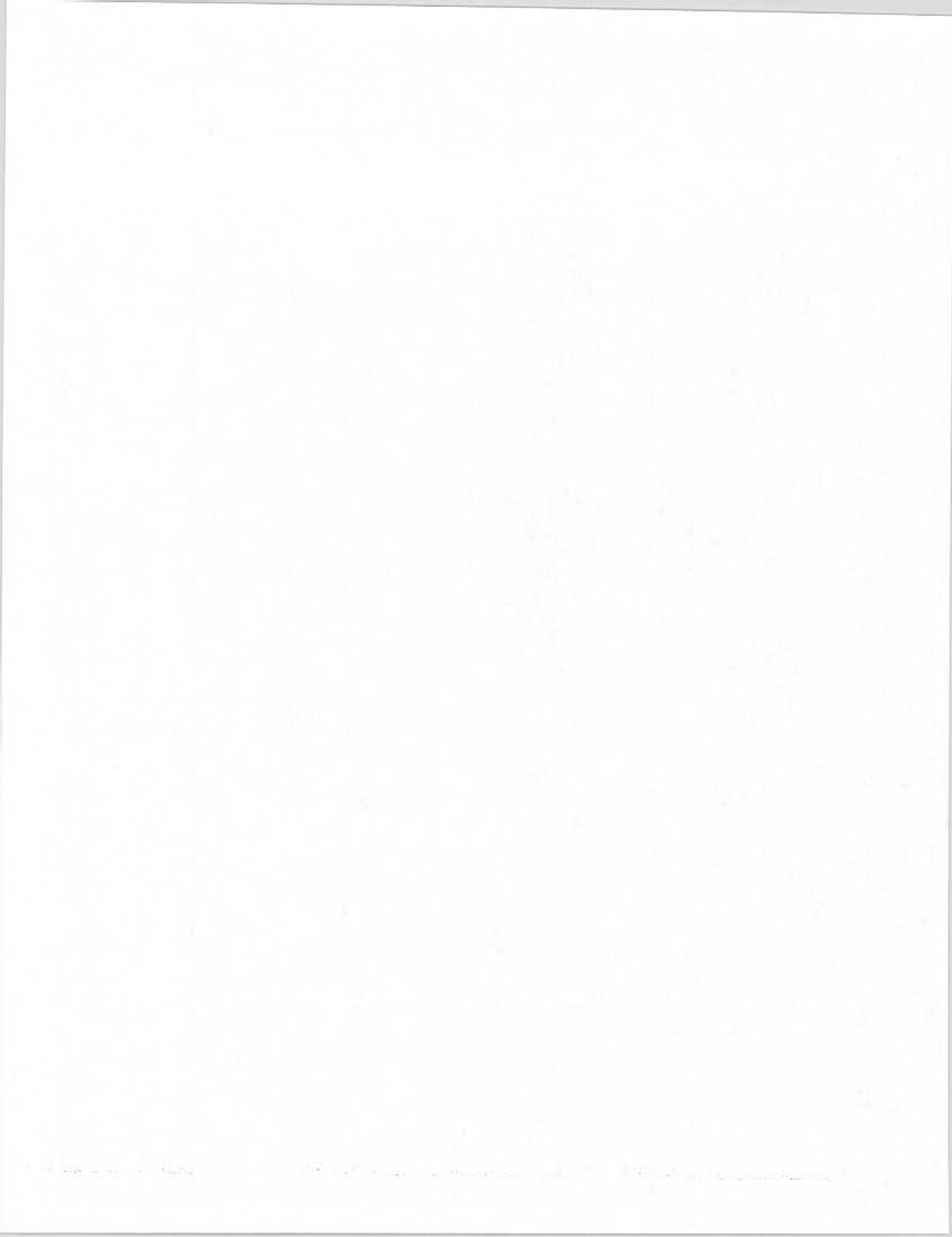
SOFTWARE DELIVERY DOCUMENTATION

This document describes the media and format on which the source code and object code of the CPCI titled "Digital Wind Sensor Application Program", CPIN 83M-FMQ13-F002-00A is delivered. Both types of code are delivered on a 5.25 inch, double sided, double density floppy disk with a formated capacity of 360K bytes.

The source code resides on one floppy disk. The disk is formated into nine sectors per track with 40 tracks per side, yielding a total of 80 tracks. Each record is 512 bytes. Source code is accessed by block number using the target compiler disk "Digital Wind Measuring System Target Compiler", CPIN 83M-FMQ13-F003-00A.

The object code resides one floppy disk. The disk format is nine sectors per track with 40 tracks per side, yielding a total of 80 tracks. The record is MS-DOS compatible. The object code is provided in three different formats in three separate files. The file "SENSOR.BIN" is a memory image of the program in binary format. The file "SENSOR.HEX" provides the object code in Intel Hex Format. The file "SENSOR.ASC" is an ASCII listing of the object code.

These floppies should be protected from dust and contamination. The floppies should not be folded or otherwise deformed. Care should be taken to avoid exposure to intense magnetic fields.



McClellan AFB
Contract #: F04606-85-C0733
CDRL Item: E004
24 May 90

VERSION DESCRIPTION DOCUMENT
FOR THE
AN/FMQ-13(V)
DIGITAL WIND SENSOR
APPLICATION PROGRAM

CPIN: 83M-FMQ13-F002-00A
ENGR SPEC #: MMA 81-025B
REVISION: INITIAL RELEASE

1. INVENTORY

The following table provides a list all items covered by the CPCl number and version. In addition all utility and/or support computer program release documents which are not part of the released item but which are required to operate, load, or regenerate the released CPCl are listed with the quantity marked as "REF".

INVENTORY LIST

ITEM	QTY	DESCRIPTION
1	2	DISK, 5.25" FLOPPY, PROGRAM SOURCE CODE
2	1	DISK, 5.25" FLOPPY, PROGRAM OBJECT CODE
3	1	LISTING, PAPER COPY, SOURCE CODE
4	1	LISTING, PAPER COPY, OBJECT CODE
5	3	SPECIFICATION, CPCl
6	3	DOCUMENT, SOFTWARE DELIVERY
7	3	DOCUMENT, VERSION DESCRIPTION
8	REF	COMPILER, "DIGITAL WIND MEASURING SYSTEM TARGET COMPILER" CPIN 83M-FMQ13-F003-00A
9	REF	MANUAL, "polyFORTH II REFERENCE MANUAL"
10	REF	MANUAL, "INTEL 8086 CPU Supplement to the polyFORTH II Reference Manual"

2. INVENTORY OF CPCl CONTENTS

The CPCl being released is titled " Digital Wind Sensor Application Program", CPIN 83M-FMQ13-F002-00A. This is the operational program that runs the ML-660/FMQ-13(V) Standard Wind Direction and Speed Sensor and the ML-660A/FMQ-13(V) Rugged Wind Direction and Speed Sensor. This program resides in one 128Kbit (16K by 8 bits) EPROM. A description of the program, source code and object code listings, and charts of the program are documented in CPCl specification titled "Computer Program Specification for the AN/FMQ-13(V) Digital Wind Sensor Application Program" supplied with this release.

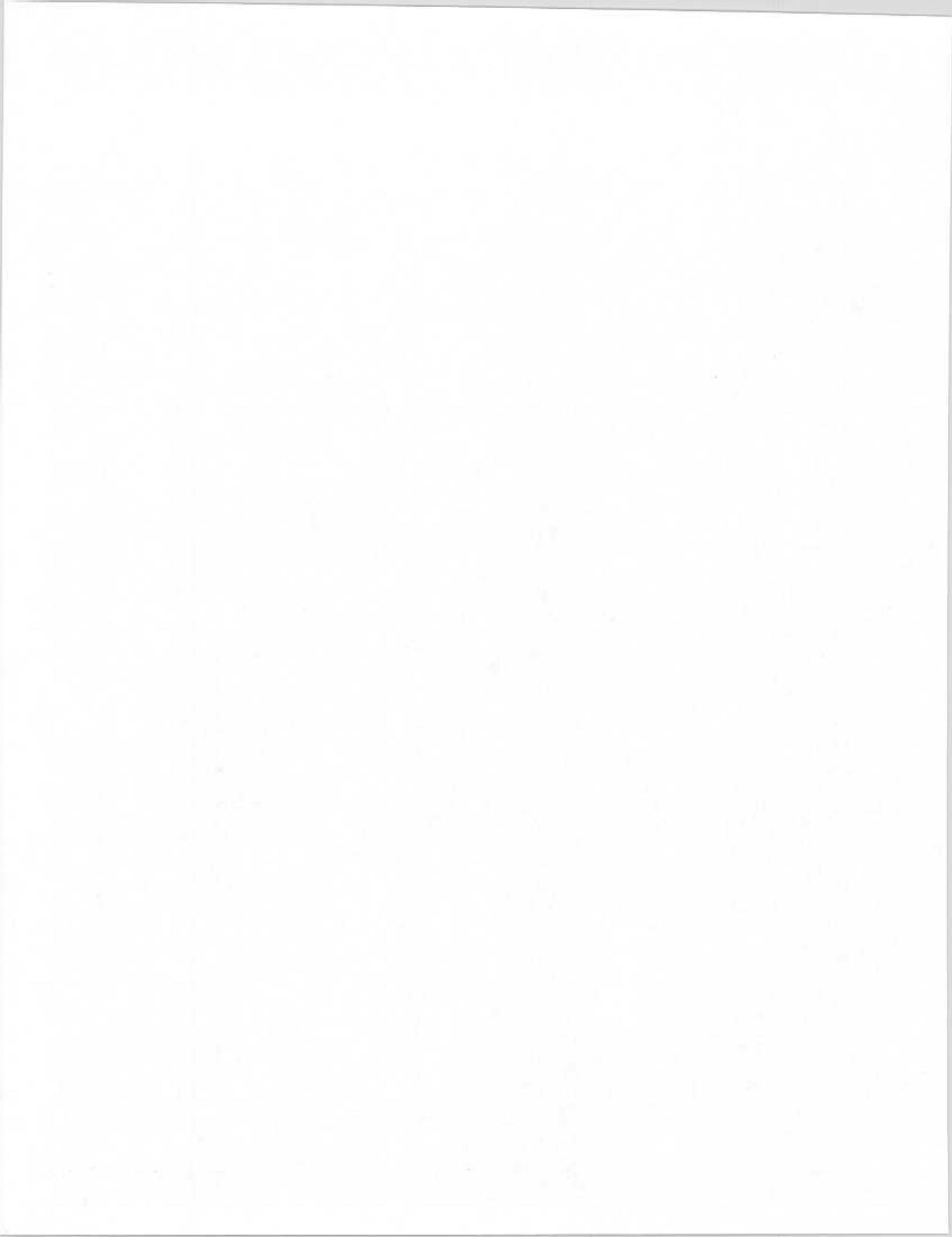
The object code is provided in three different formats in three separate files. The file "SENSOR.BIN" is a memory image of the program in binary format. The file "SENSOR.HEX" provides the object code in Intel Hex Format. The file "SENSOR.ACS" is an ASCII listing of the object code. A paper copy listing of the Intel Hex format and the ASCII format is provided with this release.

3. CLASS II CHANGES INSTALLED

This is the initial release of this program. As such it has no class II changes.

4. CLASS I CHANGES INSTALLED

This is the initial release of this program. As such it has no class I changes.



5. ADAPTATION DATA

This program does not require any adaptation data.

6. INSTALLATION INSTRUCTIONS

This is the operational program that runs the ML-660/FMQ-13(V) Standard Wind Direction and Speed Sensor and the ML-660A/FMQ-13(V) Rugged WInd Direction and Speed Sensor. This program resides in one 128Kbit (16K by 8 bits) EPROM. The program is delivered with each sensor as firmware. This EPROM resides on a PCBA referred to as the Micro-B board (P/N 66733ASSY6461-1041). The EPROM is not a field replaceable item. Due to the unique characterization data conteained in the EPROM of each sensor the EPROMs are not interchangeable with other sensors. If required for depot maintenance a new EPROM may be generated by re-characterizing the sensor in a wind tunnel using approved test procedures. These test procedures can be found in Section II of Chapter 5 of the Maintenance Manual TO-31M1-FMQ13-2. The program code for each sensor is the same. The data tables for each sensor are unique. Equipment for collecting the chacterization data and programing the EPROM data tables is provided as ground support equipment.

McClellan AFB
Contract #: F04606-85-C0733
CDRL Item: E00D
15 May 90

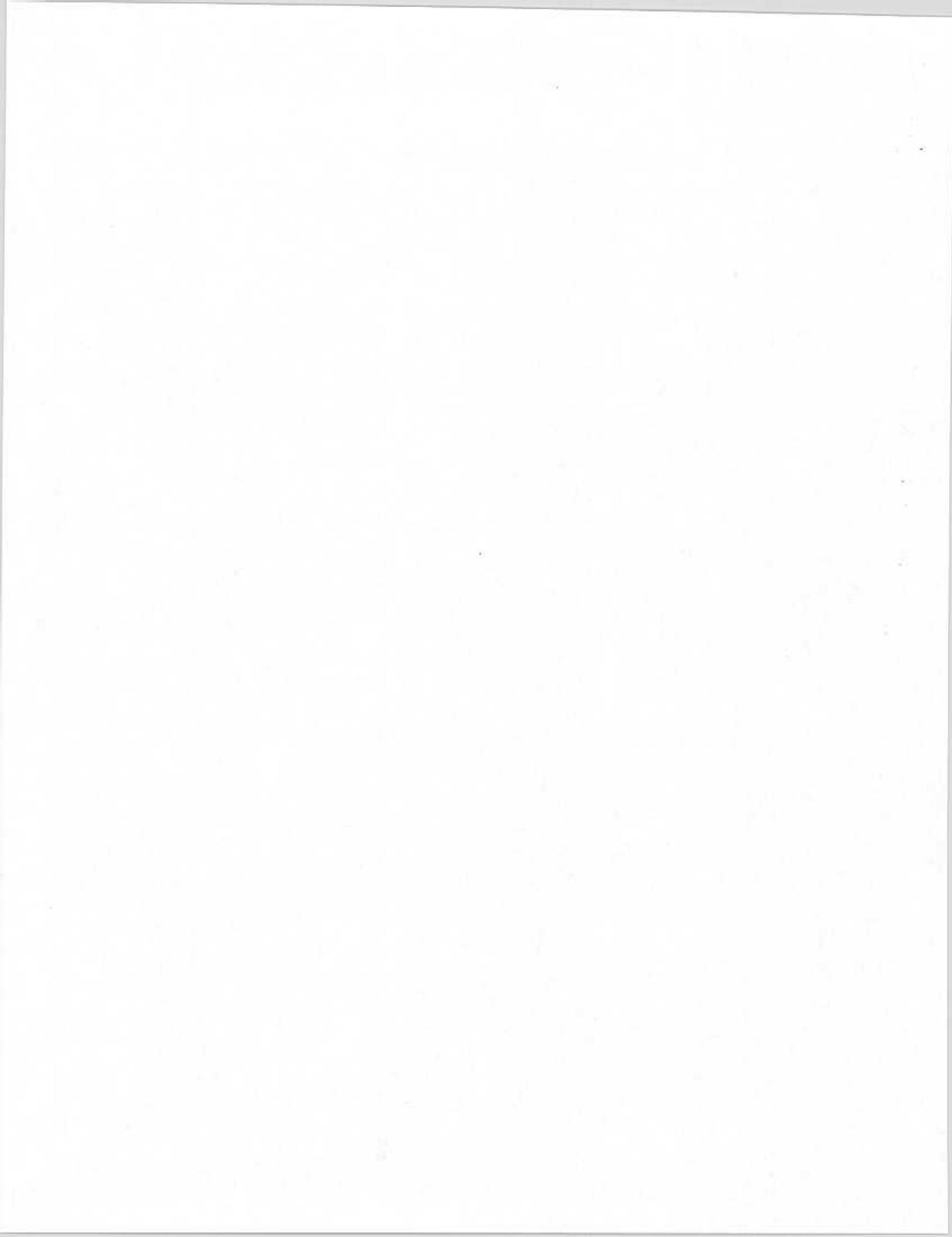
COMPUTER PROGRAM SPECIFICATION
FOR THE
AN/FMQ-13(V)
DIGITAL WIND SENSOR
APPLICATION PROGRAM

AUTHENTICATED BY _____ APPROVED BY _____
(PROCURING ACTIVITY) (CONTRACTOR)
DATE _____ DATE _____

TABLE OF CONTENTS

1. SCOPE	1
2. APPLICABLE DOCUMENTS	1
3. REQUIREMENTS	2
3.0 GENERAL DESCRIPTION	2
3.0.2 Sensor Theory of Operation	3
3.0.3 Sensor Hardware	5
3.0.3.1 Element Driver Subsystem	5
3.0.3.2 Barometric Subsystem	6
3.0.3.3 Measurement Subsystem	6
3.0.3.4 Communication Subsystem	6
3.0.3.5 Control/Status Subsystem	6
3.0.4 Inter-Assembly Communications	7
3.0.5 Programming Language	7
3.1 CPCI STRUCTURAL DESCRIPTION	7
3.2 FUNCTIONAL FLOW DIAGRAMS/CHARTS	8
3.2.1 Control Functional Flow Diagram	8
3.2.2 Data Flow Diagram	9
3.3 INTERFACES	9
3.3.1 The Sensor Microprocessor System	9
3.3.1.1 Inter-Assembly Communication Interface	9
3.3.1.2 Analog Data Input/Output Interface	10
3.3.1.2.1 Data Acquisition System Interface	10
3.3.1.2.2 Digital to Analog Interface	10
3.3.1.3 Programmable Timer Interface	10
3.3.1.4 Watch Dog Timer Interface	12
3.3.2 Inter-Assembly Communication Protocol	12
3.3.2.1 Inter-Assembly Communication Message Format	14
3.3.2.1.1 Inter-Assembly Communication Opcode	14
3.3.2.1.2 Inter-Assembly Communication Message Data	14
3.3.2.1.3 Inter-Assembly Communication Message Timing	14
3.4 PROGRAM INTERRUPTS	15
3.4.1 NMI Interrupt	15
3.4.2 Inter-Assembly Receiver Interrupt	15
3.4.3 Inter-Assembly Communication Transmitter Interrupt	15
3.4.4 Spurious Interrupt	15
3.5 TIMING AND SEQUENCING DESCRIPTION	17
3.6 SPECIAL CONTROL FEATURES	17

3.7 STORAGE ALLOCATION	17
3.7.1 Data Base Definition	17
3.7.1.1 File Description	17
3.7.1.2 Table Description	17
3.7.1.3 Item Description	17
3.7.1.3.1 Configuration Constants	17
3.7.1.3.2 Element Resistance Table	17
3.7.1.3.3 Gain and Offset Table	18
3.7.1.3.4 Temperature Table	18
3.7.1.3.5 Pressure Table	18
3.7.1.3.6 X&Y Power Difference Tables	18
3.7.1.3.7 X&Y Raw Velocity Tables	18
3.7.1.3.8 Angle Index Table	18
3.7.1.3.9 Speed Index Table	18
3.7.1.3.10 Magnitude and Angle Compensation Tables	18
3.7.1.4 Graphic Table Description	19
3.7.1.5 CPC1 Constants	19
3.7.2 CPC Relationship	19
3.8 OBJECT CODE CREATION	19
3.9 ADAPTATION DATA	19
3.10 DETAIL DESIGN DESCRIPTION	21
3.10.1 Identification of CPC No. 1	21
3.10.1.1 Detailed Description of CPC No. 1	21
3.10.2 Identification of CPC No. 2	21
3.10.2.1 Detailed Description of CPC No. 2 Figure 8	21
3.10.2.2 Detailed Description of CPC No. 1 Figure 9	22
3.10.2.3 Detailed Description of CPC No. 1 Figure 10	22
3.10.2.4 Detailed Description of CPC No. 1 Figure 11	22
3.10.3 Identification of CPC No. 3	22
3.10.3.1 Detailed Description of the Inter-Assembly Task	23
3.10.3.1.1 Description of Inter-Assembly Task Figure 12	23
3.10.3.2 Detailed Description of the Inter-Assembly Communication Interrupt Handler	23
3.10.3.2.1 Interrupt by Carrier Status Change	23
3.10.3.2.2 Transmit Buffer Empty Interrupt	23
3.10.3.2.3 Received Character Interrupt	24
3.10.4 Identification of CPC No. 4	24
3.10.4.1 CPC No. 4 Description	24
3.10.4.1.1 CPC No. 4 Description of the CPU Test	25
3.10.4.1.2 CPC No. 4 Description of the ROM Test	25
3.10.4.1.3 CPC No. 4 Description of the RAM Test	25
3.10.4.1.4 CPC No. 4 Description of Inter-Assembly Communication Loop Test	25
3.10.5 Identification of CPC No. 5	25
3.10.5.1 Description of CPC No. 5	26
3.11 PROGRAM LISTINGS	26
3.11.1 Naming Conventions	26
4. QUALITY ASSURANCE	26
4.1 TEST PLAN/PROCEDURE CROSS REFERENCE INDEX	27
5. PREPARATION FOR DELIVERY	27



TABLES

TABLE I ALLOCATION OF FUNCTION MATRIX	8
TABLE II AD/MX/MSB BIT ASSIGNMENTS	11
TABLE III AD/CSR BIT ASSIGNMENTS	11
TABLE IV INTER-ASSEMBLY OPCODE BIT	13
TABLE V INTER-ASSEMBLY OPCODES	13
TABLE VI INTERRUPTS	16
TABLE VII CONSTANTS	20

1. SCOPE

This specification establishes the requirements for complete identification of the Digital Wind Sensor Application Program (CPIN 83M-FMQ13-F002-00A) to be formally accepted by the procuring activity. This CPCI is the operation program for the ML-660/FMQ-13(V) Standard Wind Direction and Speed Sensor ML-660A/FMQ-13(V) Rugged Wind Direction and Speed Sensor, referred to hereafter as the sensor. This documentation was prepared in accordance with Data Item Description DI-E-30145 and MIL-STD-483. The program was developed in accordance with MIL-STD-1679 to the extent specified in paragraph 3.3.11 of Engineering Performance Specification MMA-81-025B.

2. APPLICABLE DOCUMENTS

Specifications

Other specifications

MMA 81-025B	22 Apr 85	Engineering Performance Specification for a Wind Measuring System
-------------	-----------	---

STANDARDS

Federal

FED-STD-1010	11 Aug 77	Telecommunications: Bit Sequencing of American National Standard Code for Information Interchange (ASCII) in Serial by Bit Data Transmission
--------------	-----------	--

FED-STD-1011	11 Aug 77	Telecommunications: Character Structure and Character Parity Sense for Serial by Bit Data Communication in the American National Standard Code for Information Interchange (ASCII)
--------------	-----------	--

FED-STD-1020	24 Sep 75	Telecommunications: Electrical Characteristics of Balanced Voltage Digital Interface Circuits
--------------	-----------	---

FED-STD-1030A	31 Jan 80	Telecommunication: Electrical Characteristics of Unbalanced Voltage Digital Interface Circuits
---------------	-----------	--

Military

MIL-STD-129H Thru Notice 2	3 Jan 78 1 Jul 80	Marking for Shipment and Storage
-------------------------------	----------------------	----------------------------------

MIL-STD-130F	21 Apr 82	Identification Marking of US Military Property
MIL-STD-143B	12 Nov 69	Standards and Specifications, Order of Preference for the Selection of Military Communication System Thru Technical Standards
MIL-STD-188C Notice 2	24 Nov 69 17 Nov 76	
MIL-STD-188-114	24 Mar 76	Electrical Characteristics of Digital Interface Circuits
MIL-STD-188-120	15 May 76	Military Communications System Standards and Definitions
MIL-STD-470	21 Mar 66	Maintainability Program Requirements
MIL-STD-483	21 Mar 79	Configuration Management Practices for Systems, Equipment, Munitions, and Computer Programs
MIL-STD-471A	27 Mar 73	Maintainability Verification/Demonstration/Evaluation
MIL-STD-1472C	10 May 84	Human Engineering Design Criteria for Military Systems, Equipment & Facilities
MIL-C-1679	1 Dec 78	Weapon System Software Development
Data Item Description		
DI-E-30145	9 May 76	Computer Software/Computer Program/Computer Data Base Configuration Item(s)

3. REQUIREMENTS

3.0 GENERAL DESCRIPTION

The Digital Wind Sensor Application Program provides sequencing, control, and data processing capabilities to: 1) sample wind speed and direction values, 2) process the wind data to obtain desired direction and speed outputs. The program also has a number of built-in tests (BITs) to assure reliable operation. An overview of the system is provided below to describe the functional relationship between the assemblies which comprise the system. A block diagram is shown in Figure 1. (Figures 1 through 29 are located in a separate bond section titled "Charts and Figures for the AN/FMQ-13(V) Digital Wind Sensor Application Program (DWG # 8200-1013)".

3.0.1 System Assemblies

The wind measuring system is comprised of three assemblies: 1) sensor, 2) indicator, and 3) recorder. A system can be comprised of these assemblies in various combinations ranging from one to four sensors and from one to sixteen indicators or recorders as specified in MMA 81-25B, paragraph 3.1.1. The sensor measures the direction and speed of the wind and provides these measurements as 5-second averages. (The indicator and recorder is controlled by its own microprocessor. The CPCI for the indicator and recorder is described in a separate specification.) The indicator displays various wind parameters which the application program computes from the sensor data. The recorder displays and prints wind parameters computed from the sensor data.

The sensor is a solid state wind measuring device which is controlled by a microprocessor which collects wind direction and speed information and processes this information into 5-second averages. These five second average values are the output of the sensor assembly. The next few paragraphs provide a brief theory of operation for the sensor.

3.0.2 Sensor Theory of Operation

The anemometer for the Air Force AN/FMQ-13(V) wind measuring set, is a thermodynamic anemometer. It uses heat transfer to measure wind velocity instead of using moving parts. This gives very high sensitivity to low speed winds and yet maintains the ruggedness to survive and measure high winds.

The sensor consists of two orthogonal, thick-film platinum element pairs surrounded by a protective cage. The elements are maintained above ambient temperature at a precise overheat. As the wind blows, an increasing amount of energy is required to maintain the elements at the elevated temperature. The energy required provides a measure of the wind speed. A microprocessor constantly monitors the energy used in heating the elements as well as the environmental parameters of temperature and pressure to compute the wind velocity.

The sensor element consists of a ceramic tube covered with a platinum film coated with glass. Two sensor elements are bonded together to make one of the two element pairs. Platinum is a stable metal with a significant temperature coefficient. The stability of the platinum is important to ensure the long term stability of the wind sensor. The glass provides a protective coating for the platinum to prevent contamination from occurring.

The pair of elements is needed for each axis in order to determine direction. As can be seen, the pair is bonded together in a way which prevents air flow between the individual elements. The windward element experiences a greater heat loss than the leeward element. This allows the direction of flow to be determined with no ambiguities.

In order to obtain the high accuracies required, both the temperature sensor and the elements are characterized in a temperature bath. These characterization data, coupled with the precision circuitry, allow measuring the temperature sensor to an accuracy of 0.05 degree C. Element temperature may be set to 0.1 degree C. Without this characterization of the elements and temperature sensor, errors in the overheat of up to 7 degrees C could result. This is due to tolerances in the manufacturing process for the elements and temperature sensors. With an overheat of 100 degrees C, this would result in a 14 percent error.

The wind speed is not only dependent on overheat and heat loss but is also dependent upon the properties of the air flowing across the sensor, namely, the air density, thermal conductivity, and heat capacity. These properties are dependent upon the atmospheric parameters of barometric pressure, temperature, and relative humidity. Of these, barometric pressure can have the greatest impact. The density of air is directly proportional to the barometric pressure therefore if the barometric pressure doubles, the density doubles. Barometric pressure changes with elevation and with the weather. The standard atmospheric pressure at sea level is 1013 millibars. At 5000 feet it is 840 millibars, a 21% difference. At 10,000 feet the pressure is 700 mb, a 43% difference. Even changes in barometric pressure due to the weather can be on the order of 2 to 3 percent. The FMQ-13(V) sensor compensates for this factor by measuring the barometric pressure and correcting the wind speed output accordingly.

The parameter with the next largest impact is temperature. Even though temperature can have an impact of up to 20 percent on the individual components, it changes the product of air density, thermal conductivity, and heat capacity by less than 3 percent. Since this change is known and based upon the ambient temperature which the sensor already measures, it is able to compensate for the changes.

The impact of relative humidity upon these variables is relatively small. At 25 degrees C a change in relative humidity from 0% to 100% affects their product by less than 0.3 percent and at 35 degrees C it is only 0.45 percent. Thus, humidity's effect is minimal and is presently ignored.

To completely compensate for these effects and to compensate for the individuality of each sensor, a microprocessor is used. The microprocessor makes it possible to obtain the high accuracy levels the sensor achieves and it gives a large degree of freedom in the operation of the sensor. The FMQ-13(V) sensor provides a serial digital output which means that there are no errors in accuracy introduced by the device reading the sensor.

3.0.3 Sensor Hardware

Figure 3-1 is an block diagram of the wind sensor circuitry. The element driver's are critical to the level of performance achieved with this technology. They will be described in detail first.

3.0.3.1 Element Driver Subsystem

The circuitry used to drive the elements (Figure 3-2) provides a means to both maintain an element at a set temperature and to sense the power required to maintain the temperature. Since the element has a significant temperature coefficient to its resistance, the temperature of the element can be determined from its resistance. Therefore in order to maintain a set temperature, one only needs to maintain a set resistance. The power applied to the element is V^2/R where V is the voltage across the element and R is the resistance of the element. Since platinum has a positive temperature coefficient, increasing the voltage across the element increases the power applied to it which increases its temperature and therefore its resistance. To measure the element's resistance, we place a resistor of known value in series with it. When the voltage across the reference resistor (VR REF), its resistance (R), and the voltage across the element (V ELEMENT), are known, the element's resistance (RE) can be determined by the equation:

$$RE = \frac{V_{ELEMENT}}{(VR\ REF) / R}$$

The power amplifier in the circuit ensures that $V_{ELEMENT} = N(VR\ REF)$. N is a value determined by the microprocessor. As can be seen from Equation 2, it is important to know the element's temperature accurately, but not only is it important to know the temperature, the microprocessor has to be able to set it accurately in fine steps. This is because the two elements are physically joined as a pair. If one element is at a higher temperature than the other then there would be heat flow between the elements and the wind reading would be corrupted. The voltage multiplying circuit allows the temperature of the element to be set to within $0.1^\circ C$.

Since $V_{ELEMENT} = N(VR\ REF)$ then $RE = N(R)$. Since R is known and N is set by the microprocessor, RE is known. If $V_{ELEMENT}$ is measured then the power applied to the element is $(V_{ELEMENT})^2/RE$. The sensor enable switch is turned off whenever the microprocessor is reset. This is a failsafe protection feature to ensure that a valid value for N is always used.

The element driver circuit provides the means to set the element temperature but in order to set the overheat, the ambient temperature must be known. The temperature circuit (Figure 3-3) provides the means to measure the temperature to a resolution of $0.04^\circ C$. The temperature sensor is a positive temperature coefficient resistor. The temperature sensor driver biases the temperature sensor with a low level signal in order not to cause self-heating. It provides a VT REF output to indicate the drive level. The temperature conditioning circuit amplifies the output up to a level appropriate for the measurement section.

3.0.3.2 Barometric Subsystem

The barometric pressure is measured using a strain gage type of pressure transducer. This transducer is coupled to a instrumentation amplifier for excitation and an op amp for signal conditioning. The output of the pressure sensor is directly proportional air pressure. This output is connected to the A/D converter or the measurement subsystem described below.

3.0.3.2 Measurement Subsystem

The measurement section (Figure 3-4) provides the ability to measure voltage, frequency and time. The Data Acquisition System (DAS) consists of a multiplexer, a sample and hold, its own internal clock, its own voltage reference, and the analog to digital converter (A/D). The DAS provides the ability to measure the voltages from the element driver circuits, from the temperature circuit, and from power supply monitors. The A/D status allows the microprocessor to know when the A/D has completed its conversion. The counter provides timing interrupts (NMI) to the microprocessor. The NMI Enable switch ensures that the microprocessor receives no timing interrupts after a reset until it has configured its interrupt vectors. This is required since the state of the timing circuit is undefined on power up.

3.0.3.3 Communication Subsystem

The communications section is fairly straight forward. The microprocessor communicates to other assemblies through the communication section. The UART transforms the CPU's data into and from the 1200 BAUD serial bit stream required for interassembly communication. The DR and INTR lines interrupt the CPU upon receiving data and upon completion of the transmission of data.

3.0.3.4 Control/Status Subsystem

Figure 3-5 is the block diagram for the control/status section. The status register allows the microprocessor to determine the sensor's ID and whether the barometric pressure phase locked loop is locked. The control register is how the microprocessor sets the MUX address, enables the sensor, enables the barometric circuit and enables the timing interrupts. The control register is reset upon power up and if the watchdog timer resets the microprocessor. This ensures that the sensor elements are disabled unless the microprocessor is functioning properly.

3.0.3.5 CPU Subsystem

The CPU section shown in Figure 3-6 contains the microprocessor and the support functions it requires to function properly. These are a clock generator, memory (RAM and ROM), chip selects for both Memory and I/O space, an interrupt controller, and buffer for the I/O busses. The buffer between the CPU and I/O busses helps keep the digital noise of the CPU out of the analog circuitry of the element drivers. The watchdog timer ensures that the CPU is functioning properly. The watchdog timer requires that the CPU reset it periodically or the watchdog will reset the CPU. This ensures that the CPU is functioning properly.

3.0.4 Inter-Assembly Communications

All the assemblies communicate with each other via an inter-assembly communications network consisting of cabling between the separate units over which serial digital information is transmitted. This network forms a broadcast bus which means information from one assembly will broadcast to all other assemblies simultaneously over the same physical link. The protocol, which governs access to the network, incorporates a master control unit. This master unit, which is switch selectable and can be either an indicator or recorder, polls the sensors for the latest data every 5 seconds. When a sensor is polled it transmits the latest 5-second wind sample three (3) times. This will give each receiving assembly three (3) chances to acquire the data. Each assembly processes the data as soon as it is received. If the master unit does not receive a response or receives a corrupted response it will repoll the sensor. Sensor assemblies listen to the communications on the network and respond with an output if the identification code of the poll request matches the identification code of the sensor.

3.0.5 Programming Language

The Digital Wind Sensor Application Program is written in the programming language known as FORTH. This particular form of FORTH is called polyFORTH. One of the attributes of polyFORTH is its multi-tasking capabilities. Multi-tasking provides a means of supporting several processing tasks concurrently. This is accomplished by using a round-robin control structure which allows each programming task an opportunity to process data as the need arises. If a task does not have any current need to process data, it is by-passed as the round-robin control progresses. Multi-tasking improves the processing efficiency of the system.

3.1 CPCI STRUCTURAL DESCRIPTION

There are ~~eight~~ (5) major components of the Digital Sensor Application Program. These components are:

- 1) Analog Data Input and Output (ADIO);
- 2) Wind Data Processing (WDP);
- 3) Inter-Assembly Communications (IAC);
- 4) Built In Test (BIT);
- 5) Start-Up and Recovery Routines (STUR).

An allocation of function matrix is provided by Table I.

TABLE I
ALLOCATION OF FUNCTION MATRIX

	Analog Data <u>Input/ Output</u>	Inter- Assembly Comm <u>Task</u>	Wind Data Process <u>Task</u>	BIT Task	Startup Task	<u>Recovery</u>
Sequence Control	*					
I/O Control	*	*				
Data Processing	*		*			
Error Detection	*	*	*	*	*	
Real Time Diagnostics	*	*	*	*	*	
Fault Recovery				*		

3.2 FUNCTIONAL FLOW DIAGRAMS/CHARTS

Functional diagrams of control and data flow are provided in Figures 2 and 3 respectively.

3.2.1 Control Functional Flow Diagram

Many of the program functions are implemented at fixed time intervals. Sequencing control is provided by the counter and the NMI interrupt handler. The counter provide and interrupt every quarter second. This handler maintains counters and sets flags that enable other processes at regular fixed intervals. The Analog Data Input/Output routine is enabled every quarter second. The Wind Data Processing routine is enabled by the Analog Data Input/Output routine. The Inter-Assembly Communication routine can support interrupts from the network input/output (I/O) port. Figure 2 provides a diagram of the control relations between the various CPCs.

3.2.2 Data Flow Diagram

A diagram depicting data flow is given in Figure 3. This diagram shows the input data collected by the ADIO routine as quarter second samples from each element. The samples are the summation of four samples from each of the four elements collected every 0.25 seconds. These A/D counts are converted to power readings and then to power differences between the two element pairs. These pwere differences are used in a table lookup operation to determine the quarter second values of velocity. These velocities are scaled by a factor of 128, (i.e. 1 knot = 128). Every second the latest four quarter second velocity values are averaged to form a one second average value of wind velocity, one for each axis. These one second X and Y averages are converted to polar form of speed and angle. In this form the one second values are compensated for the magnitude and angle errors unique to each sensor and recorded as part of the characterization process during manufacture process. The magnitude is then compensated for air density and temperature. Afetr being compensated the one second average values are converted back to X and Y components. The latest one second values are then placed into a five position queue containing the latest five one second values. These five values are maintained for each axis and are used to produce the final five second average which is the sensor output.

3.3 INTERFACES

The following paragraphs describe the interfaces of the microprocessor system of the sensor. This includes the protocol used for interprocessor communication, hereafter referred to as inter-assembly communications.

3.3.1 The Sensor Microprocessor System

The sensor assembly is comprised of an 8088 microprocessor with 2 Kbytes of RAM, 16 Kbytes of EPROM, a programmable timer, a watch dog timer, and two input/output (I/O) interfaces; (1) Inter-Assembly Communication, (2) Analog Data Input/Output. Figures 4 and 5 show the memory map and I/O device register map, respectively.

3.3.1.1 Inter-Assembly Communication Interface

The inter-assembly interface is provided by the 82C52 CMOS Serial Controller Interface intergated circuit (IC). This IC provides the functions of a universal asynchroneous receiver/transmitter (UART) and baud rate generator (BRG). This interface is located in memory at location 4000(H). There are four registers in the 82C52. The receiver/transmitter register is located at 4000(H) and is referenced by the constant comm/tr. The UART control and status register (UCR and USR) sets the word length, parity, and number of stop bits and monitors the status of the receive condition error (framing, parity, and overrun) and the state of the enabled interrupt signals. The USR and UCR is referenced by the constant comm/csr and references address 4001(H). The modem control register (MCR) provides control of the modem control lines (request-to-send, data terminal ready) and allows the enabling of interrurpts. This register is refenced by the constant comm/mcr which is assigned the value of 4002(H). The modem status register (MSR) provides the status of the modem control lines. This register is referenced by the constant comm/msr which is assigned the value of 4003(H). The baud rate is set at 1200, with 8 data bits, 1 stop bit and no parity.

3.3.1.2 Analog Data Input/Output Interface

The Analog Data Input/Output Interface is comprised of two separate 12-bit interfaces: (1) the interface for the Data Acquisition System, (2) interface for the digital-to-analog converter.

3.3.1.2.1 Data Acquisition System Interface

The Data Acquisition System (DAS) is located in memory at location 8000(H) and has three registers. The first register is located at 8000(H) and is referenced by the constant ad/mx/msb. The DAS has eight input channels. The desired channel is set by writing to the multiplexer (MUX) which is interface to the first three bits of the ad/mx/msb. The remaining bits of this register are used to control several separate devices and are listed in Table II. Reading from the ad/mx/msb register returns the most significant byte (MSB) of the 12-bit input. The second register is the data register for the least significant byte (LSB) of the 12-bit input. It is located at address 8001 and is referenced by the constant ad/lsb. Reading this register returns the lower 4 bits of the 12-bit input. These bits are returned in the upper nibble position. The full 12-bits of input data are recovered by reading in the msb, moving the msb to the msb position of one of the 16 bit registers of the 8088 microprocessor, reading in the lsb into the lower byte of the same register. Shifting the combined results four bits to the right will return the 12-bit value in normal position. The third register is a control register for the DAS and a status for several miscellaneous status inputs. This register is located at address 8002 and is referenced by the constant ad/csr. Writing to this register starts the DAS conversion process. Reading this register returns the status of several lines which are defined in Table III.

3.3.1.2.2 Digital to Analog Interface

Four 12-bit digital to analog converts (DAC) are used to provide the input set point to the element driver circuits. These DACs are located in the I/O memory at 16 byte intervals starting at location 00. The first and second byte of each 16 byte interval are the MSB and LSB for the 12-bit DAC output, respectively. These locations are referenced by the constants da/msb and da/lsb, respectively. Output data are preloaded to each DAC by writing the data out to the individual DACs. The output values are loaded into the DACs and a conversion commenced by writing to the DAC control register located in I/O memory at address 40(H). All DACs are loaded and enabled at the same time when the control register, referenced as da/csr, is accessed in a write operation.

3.3.1.3 Programmable Timer Interface

Interval timing is provided by the 8254 Programmable Interval Timer IC. This IC provides three 16-bit counters each with six modes of operation. The counter is located in memory at address 6000(H). The three counter registers



TABLE II
AD/MX/MSB BIT ASSIGNMENTS
(WRITE OPERATION)

BIT#	SIGNAL NAME	FUNCTION
0	MUX0	MUX CHANNEL SELECT LINE
1	MUX1	MUX CHANNEL SELECT LINE
2	MUX2	MUX CHANNEL SELECT LINE
3	SENSOR ENABLE	ENABLE ELEMENT DRIVERS
4	--	--
5	--	--
6	NMI DISABLE*	ENABLE NMI INTERRUPTS
7	HEATER ENABLE	ENABLE THE HEATER

TABLE III
AD/CSR BIT ASSIGNMENTS

BIT#	SIGNAL NAME	FUNCTION
0	SENSOR ID 0	SENSOR ID BIT 0
1	SENSOR ID 1	SENSOR ID BIT 1
2	--	--
3	--	--
4	--	--
5	110/220	100/220 AC VOLTAGE SENSE BIT
6	--	--
7	POWER SUPPLY	POWER SUPPLY STATUS BIT

are located in consecutive locations starting at 6000(H) and are reference by the constant ee0, ee1, and ee2. The control register is located at 6003(H) and is referenced by the constant ecsr. The timer is initialized for mode 2 operation which produces a square wave output. The frequency of the square wave is determined by the input frequency to the counter and the counter start point value. The input clock frequency comes from the UART clock output which is set at 19200 Hz. The desired output frequency is 4 Hz to produce regular quarter second interrupts. The counter start point is 4800(D). The counter starts at 4800 and counts down to 2400 at a rate of one count every 52 usec, (1/19200) changes the state of the output and counts down to zero at which point the output changes state again. At zero the count point is automatically reloaded and the process repeats. An NMI interrupt is generated with each low to high transition of the counter output.

3.3.1.4 Watch Dog Timer Interface

The sensor uses a timer circuit to reset the microprocessor if the program fails to service the timer at regular intervals of approximately 0.2 seconds. The interface to this watch dog timer is a write only register located in the I/O memory at address location 50(H). This location is accessed as part of the built-in-test (BIT) routines. The timer is referenced by the constant watchdog.

3.3.2 Inter-Assembly Communication Protocol

Inter-assembly communication is accomplished using a shared multi-drop serial data link configuration. Data transmission is accomplished using Bell 202 compatible FSK techniques. Asynchronous serial data transmissions occur at a 1200 baud rate over a physical network comprised of a single pair of wires (2 wires) connected to each assembly. Each character frame consists of 1 start bit, 1 stop bit, and 8 data bits with no parity generation. This signal format is used to transmit messages between the assemblies. The general protocol uses a central control element to govern the use of the network. This controller is referred to as the master assembly. The master assembly has the responsibility of polling the sensors every five (5) seconds. There is only one MASTER assembly allowed in a system. A polling session starts every 5 seconds and commences with a poll request to sensor ID#-1. (The sensor ID#s are established by jumper setting in the sensor assembly. Sensor ID#s must be assigned starting with ID#-1 and running consecutively up to a maximum of four for inter-assembly communications to work properly.) When sensor #1 receives the poll request it transmits the latest 5-second average X and Y wind values three times. This poll response and all inter-assembly transmissions are broadcast over the network to all assemblies to receive. After the master receives the poll response from sensor #1 it polls sensor #2. This process continues until each sensor in the system is polled. The "number of sensor" (NSNR) jumpers on the indicator/recorder assembly establish how many sensors will be polled by the MASTER. There are several negative acknowledge and redundancy schemes used in the protocol which is described in detail in paragraph 3.10.2.

TABLE IV
INTER-ASSEMBLY OPCODE BIT

BIT#	DESCRIPTION
0	OPCODE BIT 0
1	OPCODE BIT 1
2	OPCODE BIT 2
3	BACK UP OPCODE BIT
4	SENSOR ID BIT 0
5	SENSOR ID BIT 1
6	ACTIVE SENSOR ID BIT 0
7	ACTIVE SENSOR ID BIT 1

TABLE V
INTER-ASSEMBLY OPCODES

- 0 - Illegal Code
- 1 - Poll
- 2 - Repoll
- 3 - Poll Response
- 4 - Repoll Response
- 5 - (unassigned)
- 6 - (unassigned)
- 7 - (unassigned)
- 8 - (unassigned)
- 9 - Backup Poll
- A - Backup Repoll
- B - Backup Repoll
- C - Backup Repoll Response
- D - (unassigned)
- E - (unassigned)
- F - Master Take Over

3.3.2.1 Inter-Assembly Communication Message Format

The protocol uses message packets to exchange information between assemblies. Each message is framed by control character pairs (DLE STX and DLE ETX) and has a 16-bit cyclic redundancy check (CRC) number affixed to the end of the message for error checking as shown here:

DLE, STX, MESSAGE, DLE, ETX, CRC, CRC

3.3.2.1.1 Inter-Assembly Communication Opcode

Each message contains an Opcode as well as data. The Opcode character has eight bits of which only four are used for the actual operation code. The other four bits are used for identification purposes. Two bits are used to indicate the active sensor number and two bits are used for the sensor ID number. These bit assignments are provided in Table IV. The format of the Opcode character and a list of the codes are provided in Table V.

3.3.2.1.2 Inter-Assembly Communication Message Data

The information contained in the data field of a poll request sent by the master assembly is a modulo 16 poll count used by all the assemblies to track the sequence of poll requests and poll responses between the master and the sensors. Use of the poll count is described further in paragraph 3.10.2.2.4. A sensor response data field consists of wind data, sensor status, and an echo of the poll count received with the poll request from the master assembly. Normally, four bytes of wind data are transmitted. However, if the data value is the same as a DLE, then a second DLE character is sent to indicate that the first DLE was a data value and not a control character (e.g. DLE, ETX or DLE, STX). This provides a "transparency" of a DLE value occurring in the data field. In worst case this could produce eight bytes of wind data values.

3.3.2.1.3 Inter-Assembly Communication Message Timing

Communications are carried out using 1200 baud as the transmit and receive rate. This works out to be 8.33 milliseconds/character. Using this rate a poll request takes 66.64 milliseconds to send the following eight bytes of information:

DLE, STX, OPCODE, POLLCNT, DLE, ETX, CRC, CRC

The data message in the poll request is the poll count (POLLCNT) which is a modulo 16 count used to track message sequencing.

A poll response from the sensor will nominally require 108.3 milliseconds to send the following message :

DLE, STX, OPCODE, POLLCNT, WD, WD, WD, WD, STATUS, DLE, ETX, CRC, CRC

WD indicates one byte of wind data. The worst case time which corresponds to eight bytes of wind data (four extra DLE characters) is equal to 150 milliseconds. Worst Case Poll Response:

Each wind data field requires a DLE to follow

DLE, ETX, OPCODE, POLLCNT, DATA1...DATA8, STAT1, STAT2, DLE, STX, CRC,
CRC

18 Bytes (8.33 ms/byte) \leq 150 ms

A timing diagram of each component of a poll request and poll response as well as a complete poll session for four sensors is shown in figure 51. This timing diagram uses the worst case time of 150 ms for the poll response and the maximum latency periods for the carrier detect off-to-on and on-to-off transition.

3.4 PROGRAM INTERRUPTS

The following paragraphs describe the program interrupts: specifically, the source, purpose, type, priority, and required response for each interrupt. The presentation will progress from highest to lowest priority. Table VI provides a list of the interrupt vector number, the RAM location of the vector, interrupt handler program name, the source of the interrupt and the function of the interrupt.

3.4.1 NMI Interrupt

An NMI interrupt occurs every quarter second if the NMI DISABLE* bit is set in the ad/mx/msb register. This interrupt is serviced by the routine named "<timer>". The routine sets some flags and increments a counter for use by the Wind Data Processing, Analog Data Input/Output and the BIT tasks. As an NMI interrupt this has the highest priority of all external interrupts.

3.4.2 Inter-Assembly Receiver Interrupt

This interrupt is generated when a character is received via the inter-assembly communication interface. This is an IRQ interrupt for the purpose of processing received characters and has the highest IRQ priority level. This interrupt is serviced by the Inter-Assembly Interrupt Handler described in Paragraph 3.10.2.3. The interrupt rate will be once every 8.33 milliseconds during receipt of messages and will nominally have a 30% duty cycle for a four-sensor system.

3.4.3 Inter-Assembly Communication Transmitter Interrupt

An IRQ type interrupt occurs each time a character is transmitted when the transmit buffer becomes empty. This is serviced by the Inter-Assembly Output Handler which transmits another character if one is available. The interrupt is second in priority and will occur at a rate of one each 8.3 milliseconds during poll response transmission periods. The frequency of the interrupt will be 120 pps. The duty cycle will be approximately 9%.

3.4.4 Spurious Interrupt

The hardware is configured for only two IRQ interrupts at vector # 31 and 32. The only other IRQ interrupt possible would appear as a spurious interrupt which would be an interrupt request initiated on the IRQ input with no subsequent acknowledge. This would vector to the # 30 vector location. The service routine increments the SPURCNT variable and returns control to the program.

TABLE VI
INTERRUPTS

VECTOR#	LOCATION	HANDLER NAME	SOURCE	FUNCTION
0	00-03 (H)	DIVERR	INTERNAL-DIVISION BY 0	SERVICE DIVISION BY 0 ERRORS
2	08-0B (H)	<timer>	8254 PROGRAMMABLE TIMER	PROVIDE QUARTER SEC. TIMER
32	80-83 (H)	INTR0	SPURIOUS INTERRUPT	COUNT SPURIOUS INTERRUPTS
33	84-87 (H)	INTR1	UART RECEIVE DATA INTERRUPT	SERVICE SERIAL INPUT DATA
34	88-8B (H)	INTR2	UART "INTR" INTERRUPT	SERVICE XMIT & CARRIER CHANGE

3.5 TIMING AND SEQUENCING DESCRIPTION

Figure 2 shows the timing and sequencing relations between the CPCs. The Timer NMI interrupt is the basis for program timing. The NMI service routine set flag which are read by the A/D task and the BIT task. The NMI service routine also increment a modulo 8 counter variable which is used by the Process task to track the quarter second phases.

3.6 SPECIAL CONTROL FEATURES

There are no special control features involved with the operation of this CPCI.

3.7 STORAGE ALLOCATION

The storage allocation for the Sensor assembly is confined to the microprocessor solid state memory devices (i.e. EPROM and RAM). Figure 5 shows the memory map of the microprocessor. The RAM is comprised of an 2K x 8 bit static memory device.

3.7.1 Data Base Definition

Figure 28 gives a graphical representation of the database for the Sesnor assembly. The following paragraphs provide a detailed description of the database. The entire database is contained in EPROM.

3.7.1.1 File Description

The are no files used in the Sensor program.

3.7.1.2 Table Description

There are fourteen data tables contained in the EPROM of the sensor program. All but one of these is used by the program. The first table is not used by the program but is used for configuration identification. This table contains the serial numbers of all the characterized devices contained on the sensor. The remaining table are described below.

3.7.1.3 Item Description

3.7.1.3.1 Configuration Constants

This table contains three values; (1) the clock frequency is the measured clock frequency recorded during manufacturing process and is used to adjust the count for the interval timer that produces the quarter second interrupts, (2) the pressure offset is the barometric pressure recorded when the sensor was characterize during tunnel testing, (3) the speed limit is to define the operational range of the sensor and is set to 100 knots for the standard sensor and 150 knots for the rugged sensor.

3.7.1.3.2 Element Resistance Table

This table contains the resistance values for the elements at ten degree intervals over the temperature range of 50 degrees to 170 degrees centigrade.

3.7.1.3.3 Gain and Offset Table

This table contains the gain and offset value for the A/D and the D/A converters.

3.7.1.3.4 Temperature Table

This table contains the voltage to temperature conversion values for the temperature sensor.

3.7.1.3.5 Pressure Table

This table contains the voltage to pressure conversion values for the pressure sensor.

3.7.1.3.6 X&Y Power Difference Tables

This table is an index table for the X-Raw Velocity Table. It contains the power differences recorded at a known speed and direction during tunnel testing of the individual sensor assembly. Each entry has a matching entry in the X-Raw Velocity Table.

3.7.1.3.7 X&Y Raw Velocity Tables

This table contains the velocity recorded during tunnel testing for the corresponding element power difference. The power difference is the measured input. The raw velocity is the corresponding output.

3.7.1.3.8 Angle Index Table

This table is used to index into the magnitude and angle compensation tables. It contains the angle recorded during tunnel testing and is set up for angle ranging from 0 to 360 degrees at five degree increments.

3.7.1.3.9 Speed Index Table

This table is used to index into the magnitude and angle compensation tables. It contains the speeds at which the sensor was characterized during tunnel testing.

3.7.1.3.10 Magnitude and Angle Compensation Tables

These tables contain the characterization results of tunnel testing which records the sensor response over a range of set speeds and direction inputs. These data are used to interpolate the final magnitude and angle output for any given input within the operational range of the sensor.

3.7.1.4 Graphic Table Description

Figure 28 provides a graphical representation of the tables within the database.

3.7.1.5 CPCI Constants

Several constants used to reference device registers were defined in the section covering interfaces. Table VII lists the remaining constants used in this CPCI.

3.7.2 CPC Relationship

The data base is accessed by the Process task programs and the Start-up routines.

3.8 OBJECT CODE CREATION

The programs of this CPCI are written in the FORTH programming language and are compiled on an IBM PC computer system (or equivalent) with 256 KB of RAM and dual floppy disks. The specific version of FORTH used is 8086/IBM PolyFORTH II by Forth, Inc. The procedures for using this programming language can be found in the "PolyFORTH II Reference Manual" and the "PolyFORTH 8086 CPU Supplement to the polyFORTH II Reference Manual". Program code is created using the "Digital Wind Measuring System Target Compiler" (SPIN 83M-FMQ13-F003-00A) supplied with this CPCI. The target compiler disk provides the FORTH operating system with its utilities for disk formating and copying, printing and editing. During operation the target compiler resides in the "A" floppy disk drive of the PC with the "Digital Wind Sensor Application Program" disk in drive "B". Code compilation is commenced by entering the following instructions; "EMPTY COMPILER LOAD IBM LOAD" followed by carriage return (CR). The compiler will list the location of each word compiler to the monitor screen. Compilation will continue for approximately five minutes and will compile word in the order define by the load screen commands defined in block 325 through 328 of the application program listing. The object code will reside in memory at end of this procedure. The following command will download the object code from memory to the COM1 serial output port; "REMOTE SEND HEX 0 C000 4000 RDUMP (CR)" This command will download 16384 bytes (4000 hex) starting at RAM location 0 and transcribing it into a load module starting at address C000(H).

3.9 ADAPTATION DATA

There is no required direct data entry for functional adaptation. All adaptation is a function of hardware configuration links and switch settings. Setting of these links is described in chapter 2 of the Maintenance Manual TO 31MI-2FMQ13-2.

TABLE VII
CONSTANTS

NAME	VALUE	FUNCTION
TABLES	C000	STARTING ADDRESS OF THE DATA TABLES
XTALFREQ	C060	ADDRESS OF THE CLOCK FREQUENCY TABLE ENTRY
PRESREF	C064	ADDRESS OF THE REFERENCE PRESSURE
SPEEDLIMIT	C066	ADDRESS OF THE SPEED LIMIT ENTRY
5VOLTS	2048	A/D VALUE OF 5.000 VOLTS
RAINLIMIT	5120	DIFFERENCE LIMIT TO DETECT RAIN DISTORTIONS

3.10 DETAIL DESIGN DESCRIPTION

This paragraph contains technical descriptions of the computer program components identified in paragraph 3.1 of this specification. The charts and instruction listings which specify the exact configuration of the "Digital Wind Sensor Application Program" are contained in drawing number 8200-1013.

3.10.1 Identification of CPC No. 1

This section describes the method of analog data acquisition and an analog data output. The routine that performs this function is called "Acquisitioning" and is identified as CPC No. 1. The Acquisistioning task includes the routines listed in blocks 365 through 374.

3.10.1.1 Detailed Description of CPC No. 1

Figure 7 provide the chart of this task. The routine monitors the variable ACAUIRE which is a flag set by the NMI interrupt service routine. If the flag is not set the routine will pause to allow other tasks to run. If the flag is set the routine clears the flag reads the four element voltages four time each. It then read the inputs from the temperature sensor, and pressure sensor air pressure reading, the pressure sensor temperature reading, and the 5 volt reference input. The variable DACFLAG is tested. If set the DACs are loaded with the lastest output values. A BIT test is perfromed to determine if the A/D is returning reasonable readings. The 5 volt input is used for this test. If the reading is between 3.5 and 8 volts the reading is assumed good. If the reading is outside these limits than the microprocessor would most likely not be running. Since the processor is running readings outside these limits will cause the BIT test flag an error. The last step of this routine is to set the varilbe PROESSFLAG which is read by Process the next CPC.

3.10.2 Identification of CPC No. 2

This section describes functions performed to process the wind data and produce the desired output data. The routine which performs this function is called "Process" and is identified as CPC No. 2. The Process Task include the routines list in blocks 380 through 427.

3.10.2.1 Detailed Description of CPC No. 2 Figure 8

Figure 8 shows the processing to determine the raw velocities. This routine is run at quarter second intervals. The A/D readings for the elements are corrected for the gain and offset error induced by the A/D. These readings which represent a voltage are squared and divided by the element resistances to form the power value for each element. the power values for the element pairs (X+, X- and Y+, Y-) are subjected to provide the power difference. These differences are then used to lookup the cooresponding velocity. The velocity values are stored in the table scaled by a factor of 128.

3.10.2.2 Detailed Description of CPC No. 1 Figure 9

Figure 9 shows the processing to compensate one second average velocities. The routine starts by setting the variable flag DACFLAG to load the DACs. The latest four quarter second values are averaged into a one second average which are then converted to polar coordinates. The angle and magnitude values are used to lookup the compensation values for both the magnitude compensation and the angle compensation. The magnitude is compensated for air density and ambient temperature. The compensated magnitude is tested to see if it is greater than the operational range for the type of sensor, standard or rugged. Over range is flag if its over the set limit found in the data tables pointed to by the constant SPEEDLIMIT. The compensated values are converted back to rectangular coordinates and saved as X and Y values. The latest five one second values are then averaged to form the next five second average output.

3.10.2.3 Detailed Description of CPC No. 1 Figure 10

Figure 10 shows the processing for setting the element temperature. The temperature reading the 5 volt voltage readings are corrected for the offset and gain errors from the A/D. The temperature voltage reading is then corrected for any variance of the power supply from the five volt level since the data in the lookup table are based on an assumed ideal five volt operation level. The compensated temperature voltage reading is then used to lookup the corresponding temperature value. The resulting temperature is checked for proper operating range as part of the BIT. This value is added to the overheat value of 100 degrees centigrade. This total is then used to calculate the output setting to the DACs for each element.

3.10.2.4 Detailed Description of CPC No. 1 Figure 11

Figure 11 shows the processing for determining the air pressure. The A/D BIT flag is tested to see if the A/D is operating properly. A test is done to see if this is the first pass after a power up in which case the sums are cleared and the flag is set for the next pass. All subsequent passes proceed by correcting the A/D reading for offset and gain errors. The corrected reading are then added to a running sum. A counter is used to sum the inputs over an eight second period. At the end of eight seconds the sums are averaged and used to lookup the value for air pressure. This value is checked for proper range as part of the BIT. If the value is good then it is save and used to compensate the one second values of speed.

3.10.3 Identification of CPC No. 3

The Inter-Assembly Communication CPC is described in this section. The routines that implement the Inter-Assembly Communication functions are the Inter-Assembly Communication Task listed in blocks 452 through 471 and the inter-assembly communication interrupt service routines "<INTR1>", "<INTR2>", and <xmit> listed in blocks 482 through 484. This CPC performs control and error checking functions to assure effective and reliable acquisition of wind sensor data to all Indicator/Recorder assemblies. The basis for reliable communication is established by the form and protocol used for data transmission described in Section 3.3.2. This is enhanced by the redundancy and error checking provided by the Inter-Assembly CPC described below.

3.10.3.1 Detailed Description of the Inter-Assembly Task

The Inter-Assembly Task is diagrammed in Figures 12 through 14. Each of the following subparagraphs will describe one figure.

3.10.3.1.1 Description of Inter-Assembly Task Figure 12

The routine starts by clearing the receive flag and buffer and then "sleeping" until it is awakened by the interrupt service routine. Once awakened the routine goes through check teh type of message, the ID# in the message and the CRC of the message. If all these test pass then the sensor send a message out which repeats the latest five second average wind values three times.

3.10.3.2 Detailed Description of the Inter-Assembly Communication Interrupt Handler

A flow chart of the interrupt handler is provided by Figures 13 and 14. This is the interrupt service routine for the Inter-Assembly serial interface. This routine receives the characters which comprise inter-assembly messages and checks for proper framing of the message (i.e. DLE, STX, DLE, ETX framing). The main branch of this routine tests the UART Status Request (USR) to determine the cause of the interrupt. The action taken depends on whether the cause was a carrier detect change, a received character, or the transmission of a character. Each of these events are described below.

3.10.3.2.1 Interrupt by Carrier Status Change

An interrupt is generated if the state of the carrier signal changes from on to off or off to on. If the source of the interrupt is a change of carriers from on to off, then a test is made to determine if an STX character was received. This would indicate the receipt of at least part of a legitimate message. If the STX flag is set, then the I/O buffer flag is set and the routine then wakes up the Inter-Assembly Communication Task (IACT). The I/O buffer flag tells the IACT that a message has been received. If the STX flag is not set, then the routine wakes up the IACT without setting the I/O buffer flag. The IACT will ignore the buffer contents and setup to receive new data.

3.10.3.2.2 Transmit Buffer Empty Interrupt

The Transmit Buffer Empty (TBRE) interrupt indicates that the UART is ready for a new character. The character counter (CTR) is tested to see if it is greater than zero, which would mean that there are more characters to be transmitted. If CTR is positive, then a new character is transmitted and CTR is decremented. If CTR is not positive, then the UART is tested to see if the transmission of the last character is complete. If transmission is complete, then the carrier is turned off and the IACT is awakened.

3.10.3.2.3 Received Character Interrupt

If a new character is received, the routine resets a 200 ms Timeout counter used by the master assembly to timeout poll responses. The majority of this portion of the routine is used to detect the proper sequence at the start of a message. If the STX flag is clear, then the DLE flag is tested. If this flag is clear, then the character is tested to see if it is a DLE character. If it is, the DLE flag must be set or the routine will return from the interrupt. Once the DLE flag is set, the next incoming character is tested for the receipt of the STX character. If it is not received, then the DLE flag is reset and the buffer is clear. If it is the STX character, then the STX flag is set. After this all incoming characters are placed in the I/O buffer, the character count is incremented, and the buffer is tested for any overflow condition. If the buffer overflows, then the IACT task is awakened.

3.10.4 Identification of CPC No. 4

The following paragraph describes the Built In Test (BIT) routines. Three BITs are separate and independent programs which test the CPU, ROM, and RAM. There are several other BITs which are part of other CPCs. These will be described in this section and cross referenced to the corresponding CPC. The method of reporting the status of these tests is also presented. The charts for the BIT routines are provided in figures 16 through 24. Program code listing for the BIT task is provided in blocks 430 through 443.

3.10.4.1 CPC No. 4 Description

The BITs are a set of self test routines which are designed to monitor assembly performance and report and detect faults which may degrade performance. These tests are run continuously as a part of normal assembly operations. Each test is repeatedly performed several times each second. If a fault condition is detected the routine sets a bit in a RAM location and the Sensor Status word. Figure 15 shows the organization of this status word. The information from this status word and a similar one for the sensor are combined to form a byte called the General Status Byte. This byte determines the value normally displayed by the FPI routine. Figure 44 shows the organization of the status byte. The following paragraph describes the test routines.

3.10.4.1.1 CPC No. 4 Description of the CPU Test

The CPU test performs a sequence of operations on the four accumulators of the 8088 microprocessor, AX, BX, CX, and DX. The results of these operations are checked for validity. Detection of any error will be recorded by setting bit 0 of the Indicator/Recorder Status Word. Figure 16 shows a flow graph of the CPU test. Hexadecimal constants are loaded into the accumulators. AX and BX are summed to form FF which is left in AX. The 55 in CX is subtracted from the FF in AX with the resulting AA being stored in CX. CX and DX are summed to form FF which is stored in DX. This result is compared with the correct result, thereby testing the accumulator and the zero flag of the CPU status register. A test is also performed on the sign flag and carry flag.

3.10.4.1.2 CPC No. 4 Description of the ROM Test

The ROM test is accomplished by performing a 16-bit summation of all the locations of the EPROM to form a checksum. This value is compared with the correct checksum value which was produced and saved in the EPROM when it was programmed. The difference between these sums will cause the setting of bit 1 of the IRSW. Figure 17 shows the flow chart of this routine.

3.10.4.1.3 CPC No. 4 Description of the RAM Test

The RAM test is accomplished by writing a test pattern to a RAM location and verifying that the same pattern is read back. This test is performed on all RAM locations. The data in each location is saved before testing and restored at the completion of the test. Interrupts are disabled to maintain complete control of RAM access during this test. For this reason only small sections of RAM are tested during each iteration. Figure 18 shows a graph of this routine. Two test patterns are used at each location, a 55 pattern and an AA pattern. Both are comprised of alternating ones and zeros and are used to detect bit locations which are no longer functioning.

3.10.4.1.4 CPC No. 4 Description of Inter-Assembly Communication Loop Test

The Inter-Assembly Communication Loop Test is performed each time an assembly sends an inter-assembly message. When the first character of a message is transmitted, the receive circuitry is enabled in order to read the character back. The received character is compared to the transmitted character. Bit 6 of the IRSW is set if the characters are different. Figures 19 and 20 is a flow chart of this test operation.

3.10.5 Identification of CPC No. 5

The Start Up and Recovery (STUR) CPC is described in this section. This routine is the entry program after a power up or reset condition. The chart for this cpc is provided in figure 50. The program code is listed in blocks 605 through 636.

3.10.5.1 Description of CPC No. 5

The Start up and Recovery routine first step is to initialize the interrupt vector and the task tables in RAM and proceeding to initialize all the hardware interface devices. The POWERFLAG variable is tested to determine if a cold start is required. If so all the RAM locations above the vector tables and below the task tables are cleared. Then particular variables are initialized. A test is performed to see if the sensor has undergone several resets within the recent passed. If so an error flag is set. The configuration jumper for sensor ID and 110/220 AC operation are read and saved. Pointers to the variable length data tables are determined and saved. The A/D status is tested. If its ok the A/D task is enabled. The elements are powered up by ramping up the temperature setting. The tasks are enabled. Then the interrupts are enabled.

3.11 PROGRAM LISTINGS

Program listings are provided in a separate bound section titled " Program Listings for the AN/FMQ-13(V) Digital Wind Sensor Application Program"

3.11.1 Naming Conventions

The following naming conventions are used throughout the program listing. Attributes for name differentiation include upper and lower case lettering and hyphenated names or names containing a period.

FORTH defined words (i.e. those words defined as part of the standard Forth programming language) use capital letters or symbols (e.g. IF, THEN, ELSE, DO, LOOP, =, 0=, !, @). These are defined in the "polyFORTH II Reference Manual". Forth assembly code language is also represented in upper case letters and may include duplicate names for some words (e.g. IF, THEN, ELSE, DO, LOOP). These are defined in the "INTEL 8086 CPU Supplement to the polyFORTH II Reference Manual".

All variable names are upper case with no hyphens or periods. Application program words (the names of routines) are upper case and use a period or hyphen in the name (e.g. GET.TIME, GET.DATE, TIMESTAMP, ?.#SENSORS). Constants used to reference hardware device registers are lower case (e.g. clock, clock/csr, comm/data). Assembly code routine names are lower case with a symbol or a period within the name (e.g. set.status, clear.status, clock>, >clock).

4. QUALITY ASSURANCE

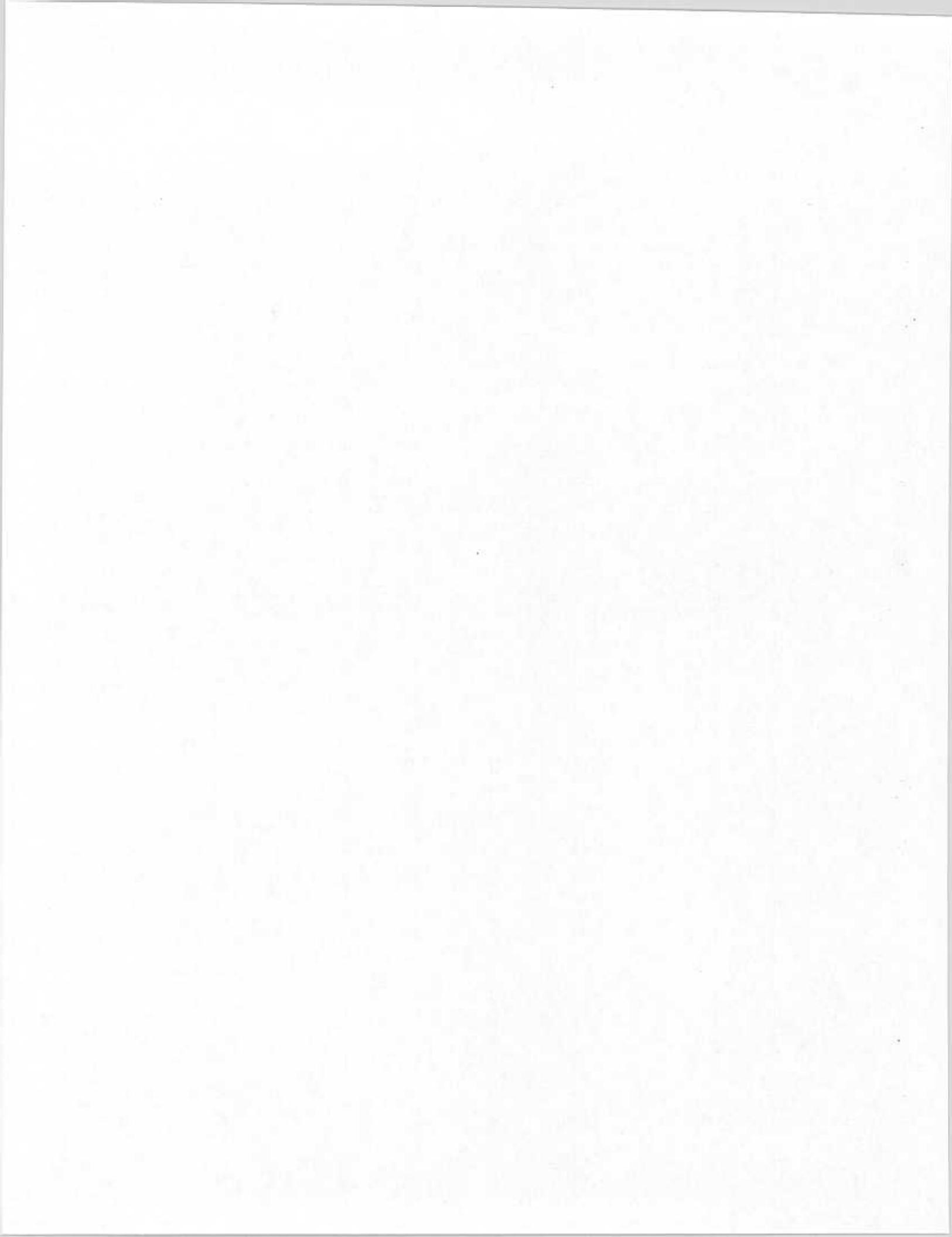
This section provides a review of the test plans and procedures used to qualify the CPC1. Qualification was performed on the built-in-test function during the maintainability demonstration. The remaining functions were qualified during performance testing.

4.1 TEST PLAN/PROCEDURE CROSS REFERENCE INDEX

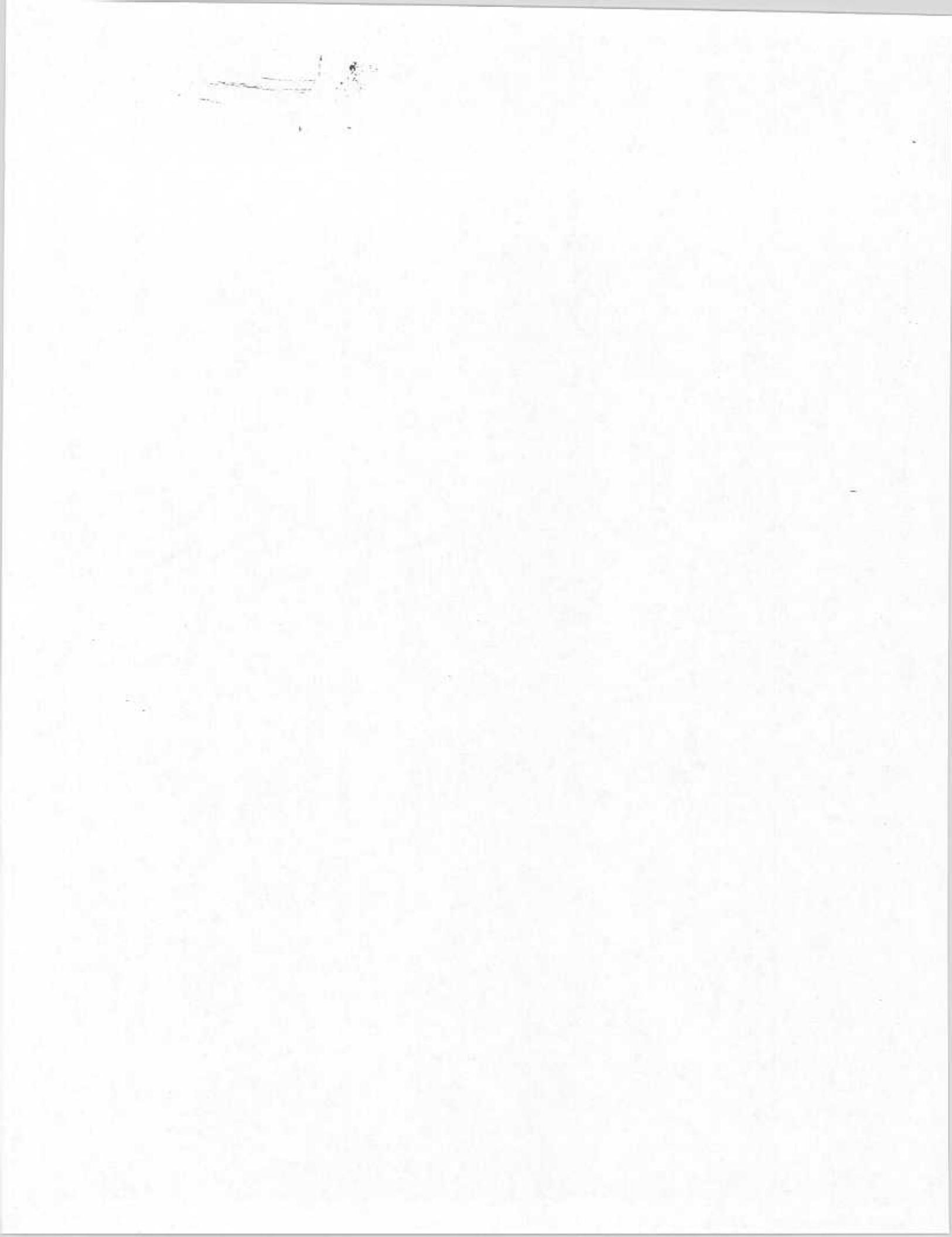
Sensor qualification tests for function were the maintainability demonstration and the performance testing. The maintainability demonstration was conducted in accordance with the Maintainability Demonstration Plan document # 9131-1008 (CDRL Item Q005) the results of which were reported in the Maintainability Demonstration Report document number WMS-MDR-01 (CDRL Item Q006). The performance test were conducted in accordance with the test procedure titled "First Article Performance Test for Wind Measuring Set AN/FMQ-13(V)" document # 9131-1005 (CDRL Item K002). This test procedure was developed in accordance with the "Equipment Test Plan for Wind Measuring Set AN/FMQ-13(V)", document # WMS-ETP-01-R1 (CDRL Item K001). Performance testing of the wind sensor envoled testing its time response to 25 knot step changes in wind speed and direction. These tests were conduct in a wind tunnel.

5. PREPARATION FOR DELIVERY

Delivery preparation for this CPCl is specified in the CDRL by items E002, E004, and E00D specifying "Software Delivery Documentation", "Version Description Document", and "Computer Software/Computer Program/Computer Data Base Configuration Item", respectively. The computer program is titled "Digital Wind Sensor Application Program". Two copies of the source code and one copy of the object code are provided on 5.25 inch diameter floppy disks wish are MS-DOS format compatible. One paper copy of the source code and object code are also specified. The disks and the paper copies will be labeled with the system nomenclature (FMQ-13(V)), the program title, release revision and the CPIN for the program.



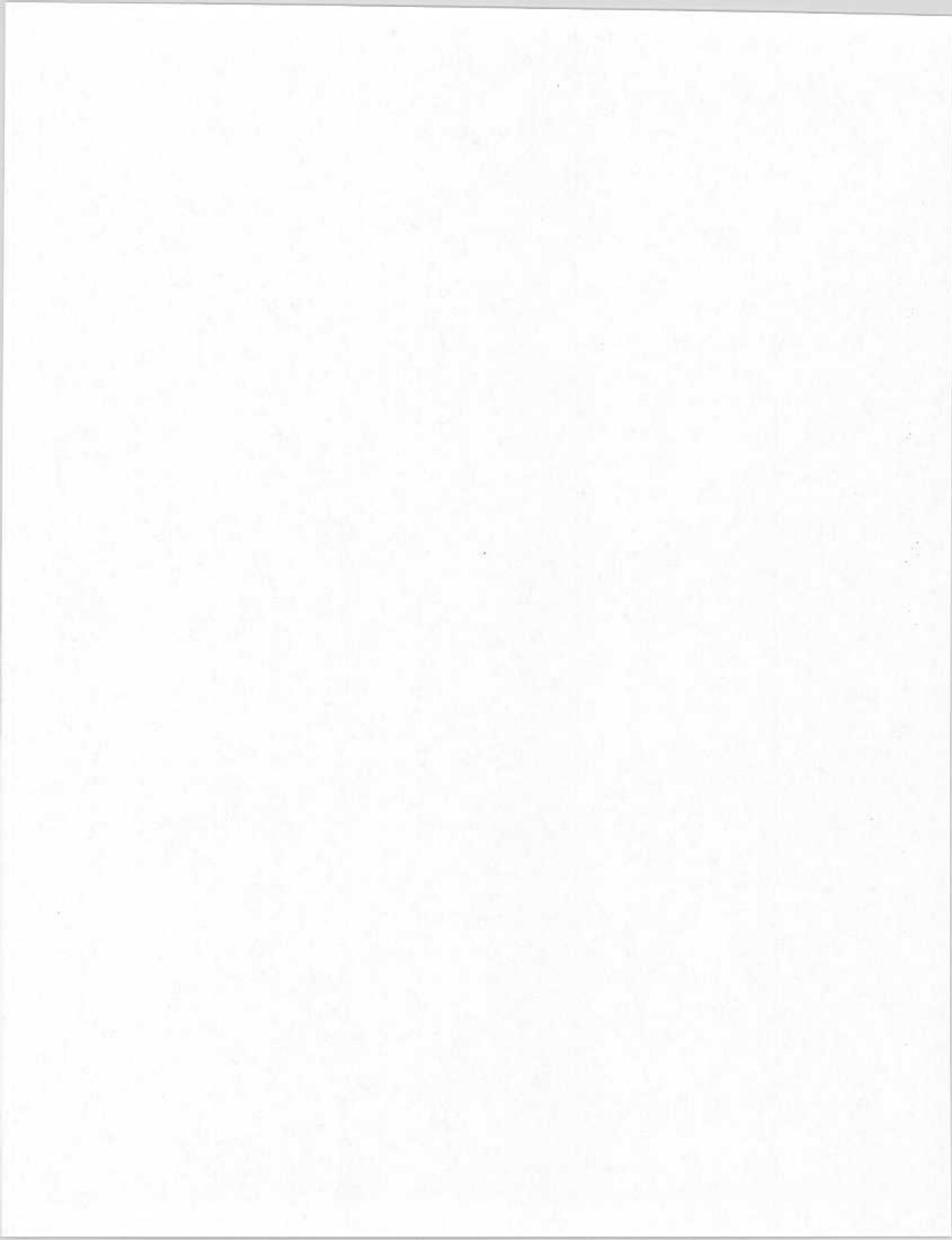
SECTION SIX
DATA SHEETS FOR THE
HARDWARE DEVICES



HS3120

DOUBLE BUFFERED

12-BIT MDAC



HS 3120 Double Buffered 12-Bit MDAC

FEATURES

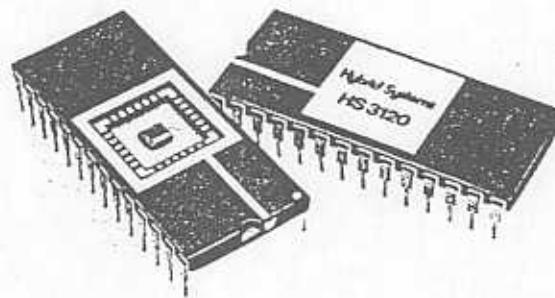
- Monolithic Construction
- 12 Bit Resolution
- 0.01% Non-Linearity
- μ P Compatible
- 4-Quadrant Multiplication
- Latch-up Protected

DESCRIPTION

The HS 3120 is a precision monolithic 12-bit multiplying DAC with internal two-stage input storage registers for easy interfacing with microprocessor busses. It is packaged in a 28-pin DIP to give high I/O design flexibility.

DOUBLE BUFFERED — The input registers are sectioned into 3 segments of 4 bits each, all individually addressable. The DAC-register, following the input registers, is a parallel 12-bit register for holding the DAC data while the input registers are updated. Only the data held in the DAC register determines the analog output value of the converter.

MICRO PROCESSOR COMPATIBLE — The HS 3120 has been designed for great flexibility in connecting to bus-oriented systems. The 12 data inputs are organized into 3 independent addressable 4-bit input registers such that the HS 3120 can be connected to either a 4, 8 or 16-bit data bus. The control logic of the HS 3120 includes chip enable and latch enable inputs for flexible memory mapping. All

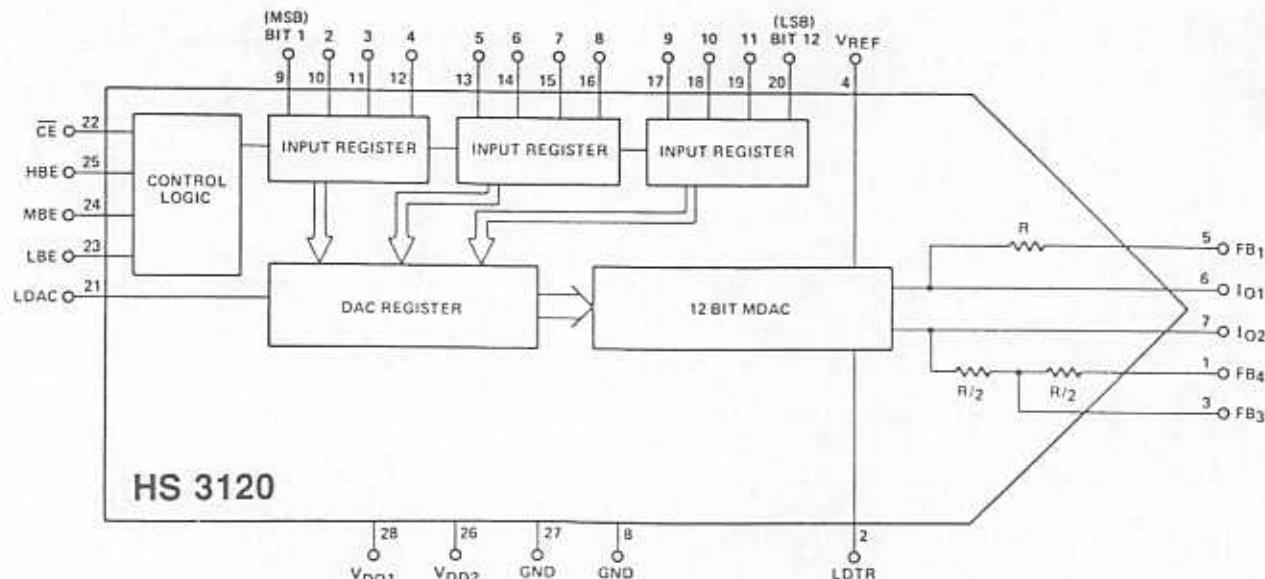


controls are level-triggered to allow static or dynamic operation.

VERSATILE OUTPUTS — A total of 5 output lines are provided by the HS 3120 to allow unipolar and bipolar output connection with a minimum of external components. The feedback resistor is internal. The resistor ladder network termination is externally available, thus eliminating an external resistor for the 1 LSB offset in bipolar mode.

MONOLITHIC CMOS CONSTRUCTION — The HS 3120 is a one-chip CMOS circuit with a resistor ladder network designed for 0.01% linearity without laser trimming. Small chip size and high manufacturing yields result in greatly reduced cost.

FUNCTIONAL DIAGRAM



SPECIFICATIONS

(Typical at 25°C , nominal power supply, $V_{\text{REF}} = +10\text{V}$, unipolar unless otherwise noted).

TYPE		HS 3120-2	HS 3120-0
DIGITAL INPUT	MULTIPLYING, DOUBLE BUFFERED INPUTS	*	*
Resolution	12-Bits		
2-Quad, Unipolar Coding	Binary ¹ , Comp. Binary ¹		
4-Quad, Bipolar Coding	Offset Binary		
Logic Compatibility ²	CMOS, TTL		
Input Current	$\pm 1\ \mu\text{A}$ (max)		
Data Set-up Time ³	250ns (min)		
Strobe Width ³	250ns (min)		
Data Hold Time ³	0ns (min)		
REFERENCE INPUT			
Voltage Range	$\pm 25\text{V}$ (max)		
Input Impedance	$8\text{k}\Omega \pm 50\%$		
ANALOG OUTPUT			
Scale Factor	$125\ \mu\text{A}/\text{V}_{\text{Ref}} \pm 50\%$		
Scale Factor Accuracy ⁴	$\pm 1\%$ F.S.R.		
Output Leakage ⁵	$<10\text{nA}$ (max) $<200\text{nA}$ (max)		
@ 25°C			
@ 125°C			
Output Capacitance			
COUT 1, all inputs high	80pF		
COUT 1, all inputs low	40pF		
COUT 2, all inputs high	40pF		
COUT 2, all inputs low	80pF		
STATIC PERFORMANCE			
Integral Linearity	$\pm 0.015\%$ F.S.R. (max)		
Differential Linearity	$\pm 0.024\%$ F.S.R. (max)		
Monotonicity	Guaranteed to 12 bits		
Monotonicity Temp. Range			Guaranteed to 10 bits
C-Models	0°C to $+70^{\circ}\text{C}$		
B-Models	-55°C to $+125^{\circ}\text{C}$		
DYNAMIC PERFORMANCE			
Digital Small Signal Settling	1.0 μsec		
Full Scale Transition Settling to 0.01% (strobed)	2.0 μsec		
Reference Feedthrough Error ($V_{\text{Ref}} = 20\text{V}_{\text{pp}}$)			
@ 1kHz	$<1\text{mV}$		
@ 10kHz	2mV		
Delay to output from Bits input	100ns ⁶		
from LDAC	200ns ⁶		
from CE	120ns ⁶		
STABILITY (Over Specified Temp. Range)			
Scale Factor ⁴	2 ppm F.S.R./ $^{\circ}\text{C}$ (max)		
Integral Linearity	1 ppm F.S.R./ $^{\circ}\text{C}$ (max)		
Differential Linearity	1 ppm F.S.R./ $^{\circ}\text{C}$ (max)		
Monotonicity Temp. Range			
C-Option	0°C to $+70^{\circ}\text{C}$		
B-Option	-55°C to $+125^{\circ}\text{C}$		
POWER SUPPLY (V_{DD})			
Operating Voltage (specifications guaranteed)	$+15\text{V} \pm 5\%$		
Maximum Voltage Range	$+5\text{V}$ to 16V		
Current	2.5mA (max)		
Rejection Ratio	0.002%/% (max)		
TEMPERATURE RANGE			
Operating C-Option	0°C to $+70^{\circ}\text{C}$		
Operating B-Option	-55°C to $+125^{\circ}\text{C}$		
Storage	-65°C to $+150^{\circ}\text{C}$		
MECHANICAL			
Case Style	28-pin double DIP		
C-Option	plastic		
B-Option	ceramic		

NOTES:

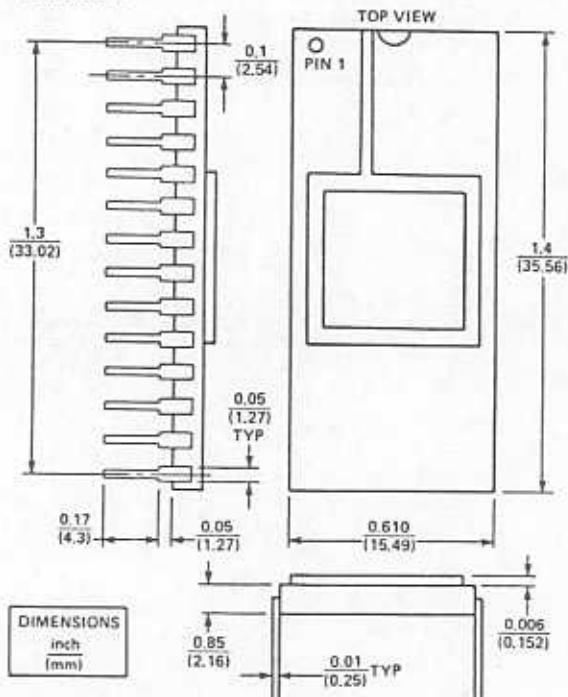
- * Same as HS 3120-2
- 1. The input coding is complementary binary if I_{O2} is used.
- 2. Digital input voltage must not exceed supply voltage or go below -0.5V.
'0' < 0.8V, 2.4V < '1' < V_{DD}.
- 3. All strobes are level triggered. See TIMING DIAGRAM.
- 4. Using the internal feedback resistor and an external opamp.
- 5. The output leakage current will create an offset voltage at the external opamps output. It doubles every 10°C temperature increase.
- 6. Delay times are twice the amount shown at T_A = $+125^{\circ}\text{C}$

PIN ASSIGNMENTS

PIN	FUNCTION
1	FB ₄ , Feedback Bipolar Operation
2	LDTR, Ladder Termination
3	FB ₃ , Feedback Bipolar Operation
4	V _{REF} , Reference Voltage Input
5	FB ₁ , Feedback, Unipolar/Bipolar
6	I _{O1} , Current out into virtual ground
7	I _{O2} , Current out-complement of I _{O1}
8	V _{SS} , Ground, Analog and DAC Register
9	Bit 1, MSB
10	Bit 2
11	Bit 3
12	Bit 4
13	Bit 5
14	Bit 6
15	Bit 7
16	Bit 8
17	Bit 9
18	Bit 10
19	Bit 11
20	Bit 12
21	LDAC, Transfers data from input to DAC register
22	CE, Chip Enable, active low
23	LBE, Bit 12 to Bit 9 Enable
24	MBE, Bit 8 to Bit 5 Enable
25	HBE, Bit 4 to Bit 1 Enable
26	V _{DD2} , Supply Analog and DAC Register
27	V _{SS1} , Ground input latches
28	V _{DD1} , Supply input latches

NOTE: Pins 8 and 27 and pins 26 and 28 must be connected externally.

MECHANICAL



CONNECTIONS

Unipolar Operation: Connect I_{O1} and FB₁ as shown in diagram. Tie I_{O2} (Pin 7), FB₃ (Pin 3), FB₄, (Pin 1) all to Ground (Pin 8)

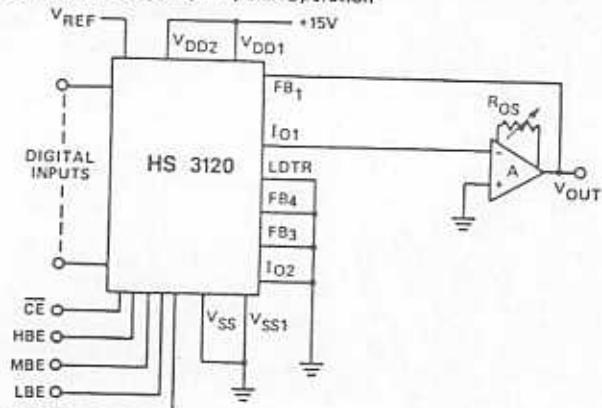
Bipolar Operation: Connect I_{O1}, I_{O2}, FB₁, FB₃, FB₄ as shown in diagram. Tie LDTR to I_{O2}

Grounding: Connect all GRD to system analog ground and tie this to digital ground.

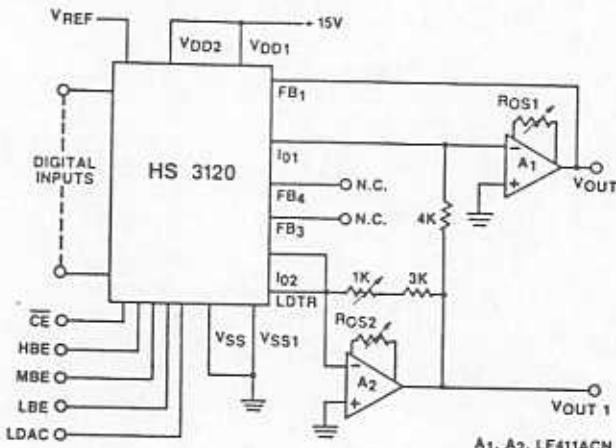
NOTE: All unused input pins must be grounded.

APPLICATIONS INFORMATION

Connection Diagram, Unipolar Operation



Connection Diagram, Bipolar Operation



Connection Diagram, Bipolar Operation (for applications where bipolar offset temperature drift ($\approx 10 \text{ ppm}/^\circ\text{C}$) is not critical)

NOTE: To maintain specified linearity, external amplifiers must be zeroed. This is best done with V_{REF} set to zero and, Unipolar: load the DAC register with all bits at zero and adjust R_{DS} for V_{OUT} = 0V

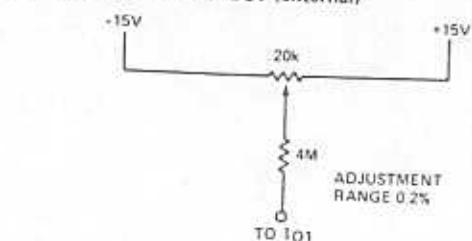
Bipolar: load the DAC register with 10...0 (MSB =1) and set R_{DS2} for V_{OUT 1} = 0V. Then set R_{DS1} for

TRANSFER FUNCTION (N=12)

BINARY INPUT	UNIPOLAR OUTPUT	BIPOLAR OUTPUT
111...111	-V _{REF} (1 - 2 ^{-N})	-V _{REF} (1 - 2 - (N - 1))
100...001	-V _{REF} (1/2 + 2 ^{-N})	-V _{REF} (2 - (N - 1))
100...000	<u>-V_{REF}</u> 2	0
011...111	-V _{REF} (1/2 - 2 ^{-N})	V _{REF} (2 - (N - 1))
000...000	0	V _{REF}

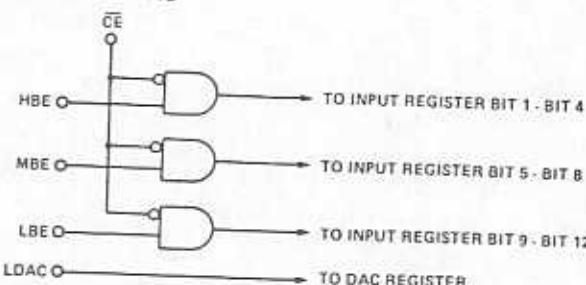


BIPOLAR OFFSET ADJUST (external)



NOTE: External opamps have to be zeroed before the bipolar offset adjust circuit is connected.

CONTROL LOGIC

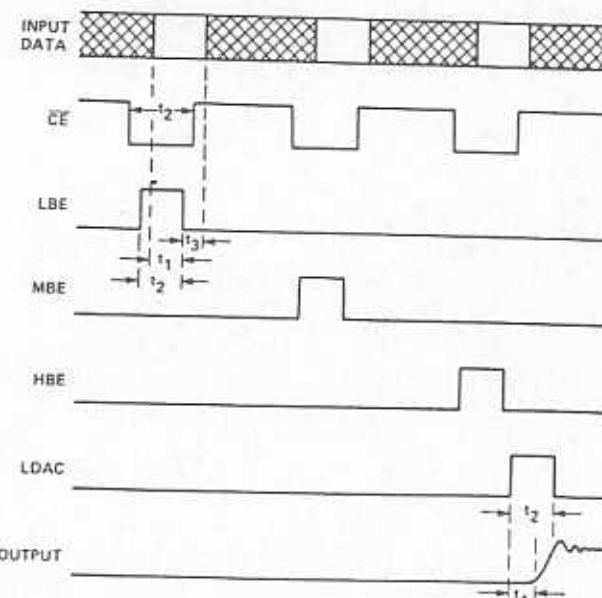


NOTE: The transfer from input register to DAC register can be performed without Enabling Chip.

STROBE LOGIC

Strobe	Function
0	data latched (held)
1	data changing (transfer)

TIMING DIAGRAM



TIME AXIS NOT TO SCALE. ALL STROBES ARE LEVEL TRIGGERED.

t₁: Data Setup Time, Time data must be stable before strobe (byte enable/LDAC) goes to "0", t₁ (min) = 250 nsec.

t₂: Strobe Width, t₂ (min) = 250 nsec. (CE, LBE, MBE, HBE, LDAC).

t₃: Hold Time, Time data must be stable after strobe goes to "0", t₃ = 0 nsec.

t₄: Delay from LDAC to Output, t₄ = 200 nsec.

NOTE: Minimum common active time for CE and any byte enable is 250 nsec.

HIGH RELIABILITY PROCESSING

Hybrid Systems maintains a continuing product assurance program to provide the highest levels of reliability demanded by MIL-specifications. Whether for government, aerospace or critical industrial applications, Hybrid Systems continues to be in a position of leadership in the development and manufacture of hybrid microelectronics data conversion products.

Facilities include the most advanced manufacturing, test and calibration equipment available, supported by Class 100 clean areas for maximum environmental control during critical assembly and inspection processes. Hybrid Systems, through its emphasis on product quality assurance and its reliability, is a proven, experienced source of converter and resistor products.

SCREENING TESTS PERFORMED PER MIL-STD-883

Process	MIL-STD-883 Test Method	Test Condition	Requirement
Pre-cap Visual	2017	-	100%
Stabilization Bake	1008	C	100%
Temperature Cycling	1010	C	100%
Constant Acceleration	2001	B	100%
Seal, Fine	1014	A	100%
Seal, Gross	1014	C	100%
Burn-in	1015	B	100%
Final Electrical Test	—	-	100%

PRODUCT SCREENING AND QUALIFICATION

Hybrid Systems is equipped to perform qualification and quality conformance testing of its products to the Level B requirements of MIL-STD-883 Rev. C. Processing to applicable Level S requirements is available where the higher confidence level is required.

ORDERING INFORMATION

MODEL	DESCRIPTION
HS 3120C-0	Double Buffered 12-Bit MDAC, Commercial
HS 3120C-2	Double Buffered 12-Bit MDAC, Commercial
HS 3120B-0	Double Buffered 12-Bit MDAC, MIL-STD-883
HS 3120B-2	Double Buffered 12-Bit MDAC, MIL-STD-883

CAUTION: ESD (Electro-Static Discharge) sensitive device. Permanent damage may occur when unconnected devices are subjected to high energy electrostatic fields. Unused devices must be stored in conductive foam or shunts. Protective foam should be discharged to the destination socket before devices are removed. Devices should be handled at static safe workstations only. Unused digital inputs must be grounded or tied to the logic supply voltage. Unless otherwise noted, the supply voltage at any digital input should never exceed the supply voltage by more than 0.5 volts or go below -0.5 volts. If this condition cannot be maintained, limit input current on digital inputs by using series resistors or contact Hybrid Systems for technical assistance.

Specifications subject to change without notice.

HS 9410 Series

8 Channel, 12-Bit Data Acquisition System with μ P Interface

FEATURES

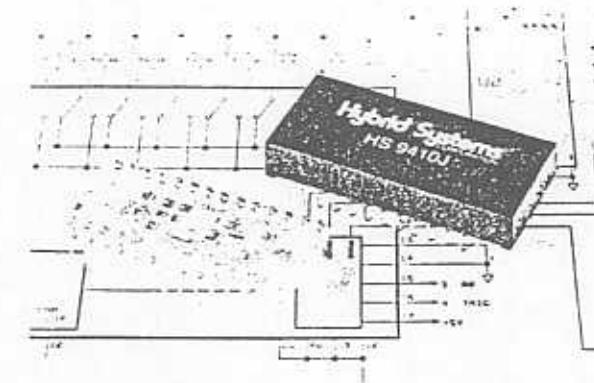
- Complete 8 Channel, 12-bit Data Acquisition System with MUX, S/H, REF, Clock and three-state outputs
- Full 8- or 16-Bit Microprocessor Bus Interface
- Guaranteed Linearity Over Temperature
- High Throughput Rate: 25kHz
- Hermetic 28-Pin Ceramic or Low Cost Epoxy DIP
- Low Power: 600mW

DESCRIPTION

The HS 9410 Series is a complete 8 channel, microprocessor compatible, 12-Bit data acquisition system with all the interface logic to connect directly to 8- or 16-Bit microprocessor buses. It is contained in a 28-pin DIP and includes an 8 channel multiplexer, a sample-and-hold amplifier, and a 12-Bit A/D converter along with the control logic needed to perform a complete data acquisition function. System throughput rate is 25 kHz for full rated accuracy.

The Analog-to-Digital converter section contains the now standard HS 574 12-Bit ADC. This ADC is implemented with advanced bipolar and CMOS LSI chips resulting in maximum performance at lowest cost. The SAR, 12-Bit decoded D/A, control logic, switches and buffers are fabricated using CMOS processing for lowest power. A unique comparator, reference and required amplifiers are fabricated using linear bipolar processes for maximum speed and reduced offset and drift over temperature.

Incorporating a unique precision comparator design, the ADC offers several advantages over more conventional circuits. A proprietary decoded 12-Bit D/A provides in-



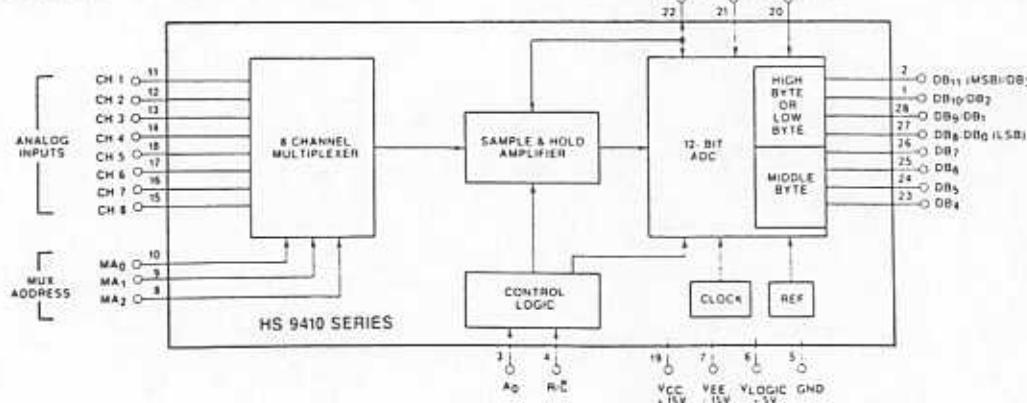
creased accuracy, lower drift and reduced output noise over the A/D operating range. Precision low TCR laser trimmed resistors are used in the converter for setting critical performance parameters including gain, offset, input ranges, and accuracy.

The HS 9410 Series is offered in a hermetically-sealed package for use over a wide temperature range and for MIL-STD-883 requirements. Hybrid Systems' proprietary commercial package is offered for applications not requiring the wider temperature exposure.

The HS 9410 Series operates from $\pm 15V^*$ and + 5V with a total power consumption of 600 mW. To take advantage of the 28-pin package the user must specify an input range of 0 to + 10V, $\pm 5V$ or $\pm 10V$ when ordering. Four basic product grades are available; J and K models are specified over a temperature range of 0°C to $+70^{\circ}\text{C}$ while the S and T models are specified over an extended temperature range of -55°C to $+125^{\circ}\text{C}$. Full screening to MIL-STD-883 Rev. C, Level B and processing in accordance with Method 5008.1 is available with models specified as "B."

* $\pm 12V$ operation possible; consult factory for further information.

FUNCTIONAL DIAGRAM



SPECIFICATIONS

(Typical values at $V_{REF} = +15V$, $V_{LOGIC} = +5V$, $V_{CC} = +5V$, unless otherwise specified.)

MODEL	HS 941J	HS 941XK	HS 941KS	HS 941XT
TRANSFER CHARACTERISTICS				
Resolution	12 Bits			
Number of Channels	8 Single Ended			
Throughput Rate	25 kHz			
ANALOG INPUTS				
Input Ranges ¹ (Specified as a suffix in the model number. See Ordering Guide.)				
HS 9410	0.10 to +10V			
HS 9411	±5V			
HS 9412	±10V			
Input Bias Current per Channel				
I_B 25°C	±1.0nA			
I_B -55°C to +125°C				
Input Impedance				
ON Channel	10 ¹⁰ Ω 100pf		±250nA max	
OFF Channel	10 ¹⁰ Ω 10pf			
DIGITAL INPUTS				
Logic Inputs				
R/C, A ₀				
V_{IH} min	+2.0V			
V_{IH} max	+5.5V			
V_{IL} max	+0.8V			
V_{IL} min	-0.5V			
I_{IL} max	±50μA max			
I_{IH} max	±50μA max			
Multiplexer Inputs				
V_{IL} max	+0.8V			
V_{IH} min	+2.4V			
Input Capacitance (All Digital Inputs)	5pF typ		+4.0V ²	+4.0V ²
Minimum Start Pulse				
R/C Negative	150ns			
SIGNAL DYNAMICS				
Conversion Time				
12 Bit Conversion	30μs max			
8 Bit Conversion	21μs max			
DIGITAL OUTPUTS				
Logic Outputs				
DB ₁₁ , DB ₉ , STS				
Logic 0	+0.4V max, $ I_{OL} \leq 1.6mA$			
Logic 1	+2.4V min, $ I_{OH} \leq 5mA$			
Leakage (High Z State)	±40μA typ (DB ₁₁ , DB ₉ only)			
Capacitance	5pF			
Output Code Configuration				
Unipolar	Positive True Binary			
Bipolar	Positive True Offset Binary			
POWER SUPPLY				
V_{LOGIC}	+4.5 to +5.5 Volts @ 20mA max			
V_{CC}	+13.5 to +16.5 Volts @ 25mA max			
V_{EE}	-13.5 to -16.5 Volts @ 35mA max			
Power Dissipation	800mW typ, 1.1W max		600mW typ, 1W max	600mW typ, 1W max
Rejection ³				
V_{LOGIC}	0.002%/ ^o C typ, 0.005%/ ^o C max			
V_{CC}	0.002%/ ^o C typ, 0.005%/ ^o C max			
V_{EE}	0.002%/ ^o C typ, 0.005%/ ^o C max			
ACCURACY				
Linearity Error (% of F.S.R. max)				
Offset ⁴	±0.025	±0.012	±0.025	±0.012
Unipolar (% of F.S.R. max)	±0.05			
Bipolar (% of F.S.R. max)	±0.25			
Gain ⁴ (% of F.S.R. max)	±0.3	±0.1	±0.25	±0.1
STABILITY				
Linearity (ppm/ ^o C max)	±0.5	±0.5	±2.5	±2.5
Unipolar Offset (ppm/ ^o C max)	±10	±5	±25	±20
Bipolar Offset (ppm/ ^o C max)	±25	±20	±25	±20
Gain (Scale Factor) (ppm/ ^o C max)	±50	±20	±50	±25
TEMPERATURE RANGE				
Operating	0° to +70°C			
Storage	-25°C to +85°C		-55°C to +125°C	-55°C to +125°C
PACKAGE	CASE A	CASE A	CASE B	CASE B

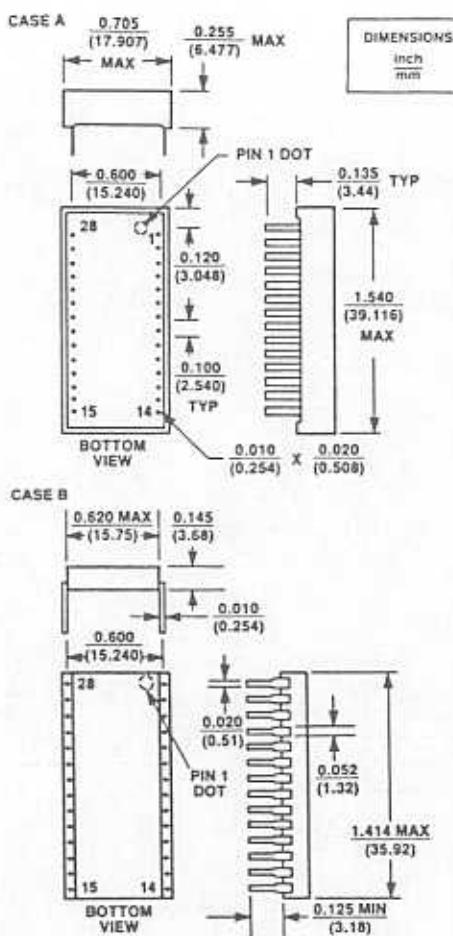
NOTES

- 1. For J and K models, positive analog input voltage should not exceed $V_{CC} = 4$ volts. Exceeding $V_{CC} = 4$ volts can cause an OFF channel to be turned ON. Negative input voltages and input voltages recommended for MA₀ / MA₂ when driven by TTL. 3. Maximum change over rated supply voltage. 4. Externally adjustable to zero. See Applications Information.

*Specifications same as HS 9410J.

PACKAGE OUTLINE

Dimensions shown in inches and (mm)



PIN ASSIGNMENTS

PIN	FUNCTION	PIN	FUNCTION
1	DB ₁₀ /DB ₂	28	DB ₉ /DB ₁
2	DB ₁₁ (MSB)/DB ₃	27	DB ₈ /DB ₀
3	A ₅	26	DB ₇
4	R/C	25	DB ₆
5	GROUND	24	DB ₅
6	V _{LOGIC}	23	DB ₄
7	V _{EE}	22	STS(STATUS)
8	MUX ADDRESS A ₂	21	GAIN
9	MUX ADDRESS A ₁	20	OFFSET
10	MUX ADDRESS A ₀	19	V _{CC}
11	INPUT CH 1	18	INPUT CH 5
12	INPUT CH 2	17	INPUT CH 6
13	INPUT CH 3	16	INPUT CH 7
14	INPUT CH 4	15	INPUT CH 8

ABSOLUTE MAXIMUM RATINGS

V _{CC} to Common GND	0 to +16.5V
V _{EE} to Common GND	0 to -16.5V
V _{LOGIC} to Common GND	0 to +7V
Control Inputs (A ₀ , R/C) to Common GND	-0.5V to V _{LOGIC} + 0.5V
Power Dissipation	1.3W
Lead Temperature, Soldering	300°C, 10Sec
Maximum Input Voltage	V _{CC} + 20V
Minimum Input Voltage	V _{EE} - 20V
Analog Input Maximum Current	25mA

CONTROL FUNCTIONS

The HS 9410 Series contains control functions necessary to provide for microprocessor interface. All control functions are defined in Tables 1, 2, and 3.

Function R/C	Definition	Function
	Read/Convert	1. L initiates conversion. 2. Low (0) disconnects data bus. 3. High (1) initiates read.
A ₀	Device Address	1. Selects conversion mode. 12-bits if low (0), 8-bits if high (1) when R/C L. 2. In read mode A ₀ selects the output format. If low (0) then 8 MSB's (high and middle byte) or if high (1) then only low byte and trailing zeroes.
MA ₀ MA ₁ MA ₂	Multiplexer Address	Select Channels 1-8 (see MUX Logic Table 3)

Table 1. Defining the Control Functions

Control Inputs		Operation
R/C	A ₀	
L	0	Initiates 12-bit conversion
L	1	Initiates 8-bit conversion
1	0	Enables 8 MSB's (high byte)
1	1	Enables 4 LSB's (low byte) and 4 trailing zeros
0	X	Output data (DB) goes to high impedance state.

Table 2. Truth Table—Control Inputs

Mux Address Inputs			Channel Selected	NOTES
A ₂	A ₁	A ₀		
0	0	0	1	1: 1 indicates logic HIGH.
0	0	1	2	2: 0 indicates logic LOW.
0	1	0	3	3: X indicates don't care.
0	1	1	4	4: L indicates operation commences on high to low transition.
1	0	0	5	5: MSB → XXXX XXXX High Byte Middle Byte
1	0	1	6	6: XXXX ← LSB Low Byte
1	1	0	7	
1	1	1	8	

Table 3. Truth Table—Multiplexer Address

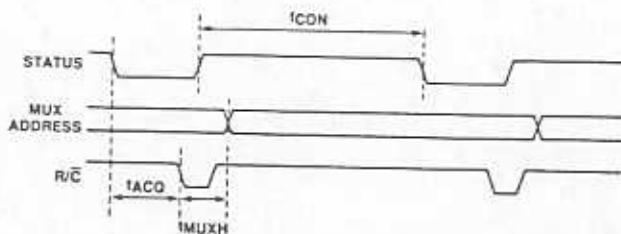
APPLICATIONS INFORMATION

TIMING

The timing diagrams are shown in Figures 1 through 6. Figures 1 and 2 show how the multiplexer addressing is related to the convert cycle, while Figures 3 and 4 show the timing sequence to start either a 12- or an 8-bit conversion. Figures 5 and 6 show how to read the multiplexed data from the internal register in the HS 9410.

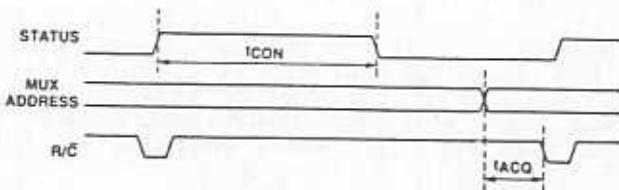
Figures 1 and 2

The multiplexer address can be changed either during or after a conversion, but care must be taken not to change the address within 1 microsecond after the convert command to insure that the sample/hold will not start to acquire the signal of the new channel. After the multiplexer address has been changed, you must allow the sample/hold at least 10 microseconds in sample mode to acquire the new input signal.



¹ MUXH	Multiplexer address hold time after convert command	1 μ s min
¹ AQO	Minimum time between conversions (S/H acquisition time)	10 μ s min
¹ CON	Conversion time for -12 bit resolution J.K-Models	30 μ s max
	S.T-Models	30 μ s max
	- 8 bit resolution J.K-Models	22 μ s max
	S.T-Models	22 μ s max

Figure 1. Timing Diagram 8/12-Bit Conversion,
MUX Address Changes During Conversion



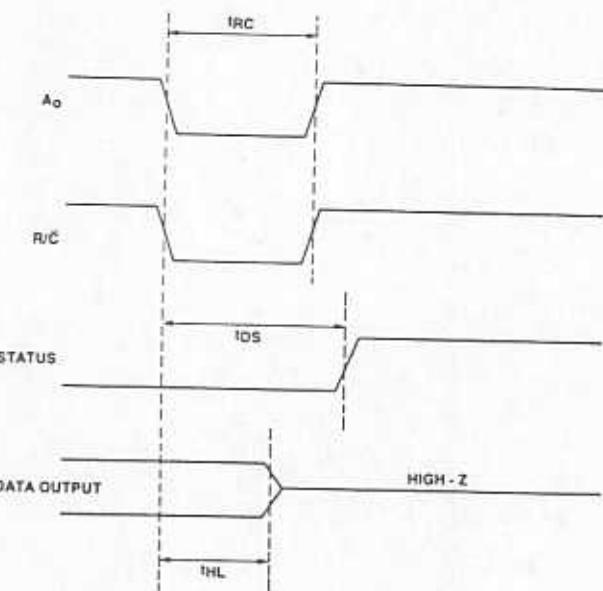
¹ AQO	Minimum time between MUX address change and convert command	
¹ CON	Conversion time - Specifications, see Figure 1	10 μ s min

Figure 2. Timing Diagram 8/12-Bit Conversion,
MUX Address Changes Between Conversions

Figures 3 and 4

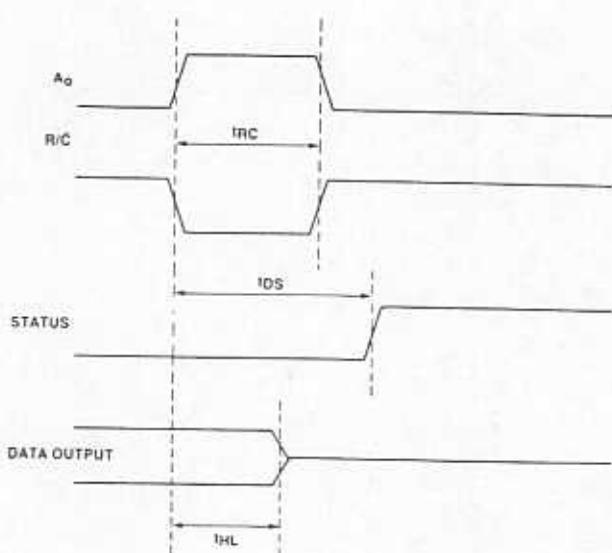
Figures 3 and 4 show how to start a convert cycle. The logic level of the A0 line determines whether a 12- or 8-bit conversion will be initiated. If A0 is low during the start convert command, a 12 bit conversion will be started; if A0 is high, an 8-bit conversion will occur. The A0 line has to be setup when the R/C line goes to logic '0' and must remain in the desired level for at least 150 ns. The R/C line is used both to start a conversion and to read the output data. If R/C is going low a con-

version is initiated. This is indicated by the STATUS line going high. A second start convert command during a conversion will be ignored. The R/C pulse must have a minimum width of 150 ns. For optimum performance the rising edge of the R/C pulse should not occur during a conversion if the conversion has been in progress for more than 1.5 microseconds, i.e., the negative R/C pulse should be either shorter than 1.5 microseconds or longer than the conversion time.



¹ RC	R/C pulse width, A0 pulse width (Note 1)	150 ns min
¹ DS	Status delay from R/C	200 ns max
¹ IHL	Output float delay	150 ns max

Figure 3. Timing Diagram to Start a 12-Bit Conversion



¹ RC	R/C pulse width, A0 pulse width (Note 1)	150 ns min
¹ DS	Status delay from R/C	200 ns max
¹ IHL	Output float delay	150 ns max

Figure 4. Timing Diagram to Start an 8-Bit Conversion

Figures 5 and 6

If a conversion is in progress the data output lines are disabled in the high impedance state. Data can be enabled by bringing the R/C line high after a conversion is complete (this is indicated by the STATUS line going low; Fig. 6). If R/C has been returned high during a conversion the data outputs will be enabled automatically after STATUS goes low (Fig. 5). The A₀ line is used to address either the 8 upper data bits or the 4 lower data bits followed by 4 trailing zeros. If an 8-bit conversion has been performed the lower 4 bits will always be '0'. After an 8-bit conversion, it is not necessary to read the lower 4 bits prior to starting a new conversion. Note that A₀ only controls the address of the two data bytes while the high impedance state of the output buffers is controlled by the R/C and STATUS line. The output buffers will not return to the high impedance state when A₀ is changed to address the second data byte.

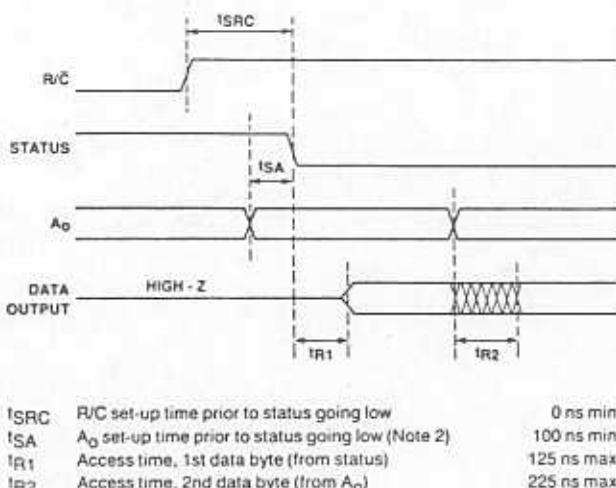


Figure 5. Timing Diagram Read Cycle,
R/C Going High During Conversion

NOTES:

- For optimum performance the positive edge of the R/C pulse should not occur during a conversion if the conversion has been started for more than 1.5 microseconds. The negative R/C pulse should be either shorter than 1.5 microseconds or longer than the conversion time.
- If the set-up time for A₀ cannot be met, the access time for the first data byte will be increased. In that case the first data byte will become valid 225 ns max after the change of the A₀ line.

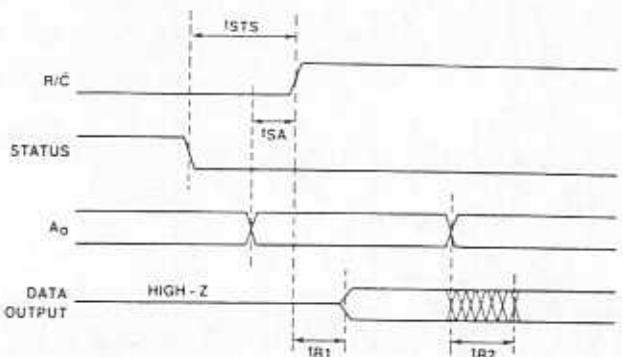


Figure 6. Timing Diagram Read Cycle,
R/C Going High After Conversion

USING THE A₀ LINE

The state of A₀ at the start of a conversion places the DAS in either a full 12-bit conversion or in an 8-bit 'short cycle' mode. During a READ at the end of a conversion A₀ is used to format the data as follows:

1. Prior to Conversion (WRITE)		MODE
A ₀ = 1		Short cycle 8-bit conversion
A ₀ = 0		Full 12-bit conversion
2. After Conversion (READ)		
A ₀ = 1		Data = Low Byte (LSB) followed by zeros
A ₀ = 0		Data = High Byte (MSB's) followed by middle byte

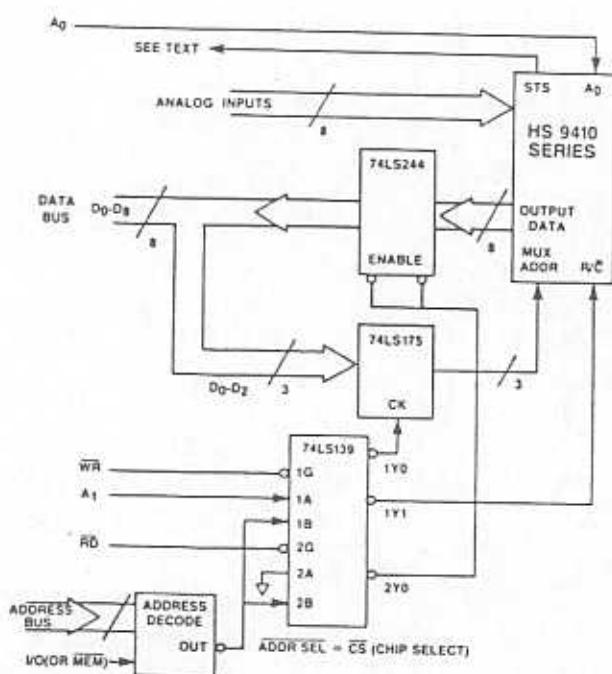
In a μP application A₀ can be considered a pair of W/R locations as follows:

1. Prior to Conversion (WRITE)		MODE
W/R = 0 in low address (A ₀ = 0)		Full 12-bit conversion
W/R = 0 in high address (A ₀ = 1)		Short cycle 8-bit conversion
2. After Conversion (READ)		
W/R = 1 in high address (A ₀ = 1)		LSB's & zeros
W/R = 1 in low address (A ₀ = 0)		8 MSB's only

MICROPROCESSOR INTERFACE

The HC 2110 Series DAS can be interfaced with most popular microprocessors. The DAS may be either positioned in a memory location (memory map) or as an I/O device. In the case of memory mapping, the DAS acts as a static RAM where READ and WRITE instructions are given to the selected address. When the DAS is connected as an I/O device, the I/O enable can be substituted for the MEMR or MEMW command. Figure 7 shows a typical scheme to implement this interface.

STS is not used in this example; the μ P must read data 30 μ s after conversion starts. This delay can be generated with NOP or other instructions inserted between the WRITE and READ functions. The STS line can also be used to cause the processor to WAIT or HALT or can be used as an interrupt line such as IREQ (in the case of 6800 or 6502).



HS 9410 Function					
A ₀	A ₁	WR	RD	ADDR SEL	Read/Write Operation
X	0	1	1	0	WRITE MUX ADDRESS
0	1	1	1	0	WRITE START 12-BIT CONV.
1	1	1	1	0	WRITE START 8-BIT CONV.
0	X	1	0	0	READ HIGH BYTE (8 MSB's)
1	X	1	0	0	READ LOW BYTE (4 LSB's)

NOTES:

1. 1 indicates logic HIGH. 2. 0 indicates logic LOW. 3. X indicates don't care.
4. \sqcap indicates operation commences on low to high transition.
5. \sqcup indicates operation commences on high to low transition.

Figure 7. Interfacing the HS 9410 Series

INPUT EXPANSION

The DAS is configured with an 8 channel high level multiplexer input. This was done to optimize package size (28 pin DIP) and cost. In the event the user wishes to increase the number of input channels, a double rank MUX input is recommended (series connected). This typical configuration is shown in Figure 8.

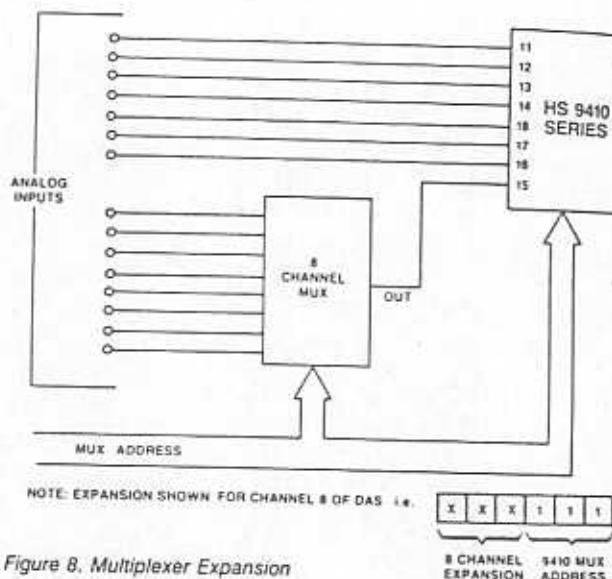


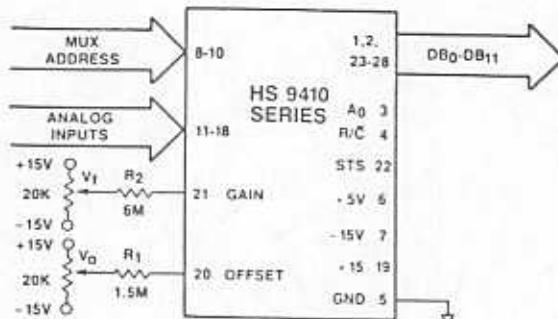
Figure 8. Multiplexer Expansion

ZERO AND GAIN CONNECTIONS

The DAS is normally used with external zero and gain calibration potentiometers. However, if maximum accuracy is not required, they may be omitted. The zero control has a range of about ± 20 LSB, and the gain control has a range of about ± 13 LSB.

Proper gain and zero calibration requires great care and the use of extremely sensitive and accurate instruments. The voltage source used as a signal input must be very stable. It also should be capable of being set to within 1/10LSB at both ends of its range.

The DAS's zero and gain adjustments are independent of each other if the zero (or offset) adjustment is made first.



NOTES:

$$1. \text{ OFFSET (VOLTS)} = V_0 \frac{5}{100 + R_1} \quad 2. \text{ GAIN (V/V)} = V_1 \frac{200}{100 + R_2}$$

Where R_1 and R_2 are shown in kilohms

Figure 9. Gain and Offset Input Connections

ZERO ADJUSTMENT PROCEDURE

1. For unipolar ranges:
 - a) Set input voltage precisely to $\pm \frac{1}{2}$ LSB.
 - b) Adjust zero control until converter is switching from 000000000000 to 000000000001.
2. For bipolar ranges:
 - a) Set input voltage precisely to $\pm \frac{1}{2}$ LSB above F.S.
 - b) Adjust zero control until converter is switching from 000000000000 to 000000000001

GAIN ADJUSTMENT PROCEDURE

1. Set input voltage precisely to $\pm \frac{1}{2}$ LSB less than 'all bits on' value. Note that this is $\pm \frac{1}{2}$ LSB less than nominal full scale.
2. Adjust gain control until converter is switching from 111111111110 to 111111111111.

Table 4 summarizes the zero and gain adjustment procedure, and shows the proper input test voltages used in calibrating the DAS.

Input Voltage Range	Adjustment	Input Voltage	Adjust input to point where converter is just on the verge of switching between the two codes shown. ¹
0 to +10V	ZERO	1.22mV	000000000000 000000000001
	GAIN	9.9963V	111111111110 111111111111
$\pm 5V$	ZERO	-4.9988V	000000000000 000000000001
	GAIN	4.9963V	111111111110 111111111111
$\pm 10V$	ZERO	-9.9976V	000000000000 000000000001
	GAIN	9.9927V	111111111110 111111111111

¹Codes shown are natural binary for unipolar input ranges and offset binary for bipolar ranges.

Table 4. Calibration Data

POWER SUPPLY CONSIDERATION

Power supplies used for the DAS should be selected for low noise operation. In particular they should be free of high frequency noise. Unstable output codes may result with noisy power sources. It is important to remember that 2.44mV is 1LSB for a 10 volt input.

Decoupling capacitors are recommended on all power supply pins located as close to the converter as possible. Suitable decoupling capacitors are 10 μ F tantalum type in parallel with 0.1 μ F disc ceramic type.

GROUNDING CONSIDERATIONS

The common at pin 5 is the ground reference point for the internal reference and is thus the high quality ground for the DAS. In order to achieve all of the high accuracy performance available from the DAS in an environment of high digital noise content, care should be taken when handling analog and digital grounds, as follows. Where analog and digital grounds are run separately on the PCB, these should be connected together at the package (pin 5). However, if the grounds are connected separately in the system for other reasons, then only the analog ground should be connected at the package to pin 5. If digital common contains high frequency noise beyond 200mV, this noise may feed through the converter, so that some caution will be required.

It is also important in the layout to carefully consider the placement of digital lines. It is recommended that digital lines not be run directly under the DAS. For optimum system performance, if space permits, a ground plane is advised under the DAS. This should be connected to a digital ground. Finally, in packaging the assembled DAS, the designer should also try to minimize any capacitive coupling that might occur at the top to the device.

ORDERING INFORMATION

Model Number ¹	Input Range	System Accuracy (% FSR)	Full Scale T.C. (ppm/ $^{\circ}$ C)	Temp. Range	MIL Screening
HS 94XXJ		± 0.025	50.0	0 $^{\circ}$ C to +70 $^{\circ}$ C	—
HS 94XXX		± 0.012	20.0	0 $^{\circ}$ C to +70 $^{\circ}$ C	—
HS 94XXS	SEE NOTE 1	± 0.025	50.0	-55 $^{\circ}$ C to +125 $^{\circ}$ C	—
HS 94XXT		± 0.012	25.0	-55 $^{\circ}$ C to +125 $^{\circ}$ C	—
HS 94XXS/B		± 0.025	50.0	-55 $^{\circ}$ C to +125 $^{\circ}$ C	883 Rev. C
HS 94XXT/B		± 0.012	25.0	-55 $^{\circ}$ C to +125 $^{\circ}$ C	883 Rev. C

NOTES:

1 HS 94XX

MODEL SUFFIX	INPUT RANGE
10	0 to +10V
11	$\pm 5V$
12	$\pm 10V$

Specifications subject to change without notice.

Add letter suffix as required above.

CAUTION: ESD (Electro-Static Discharge) sensitive device. Permanent damage may occur when unconnected devices are subjected to high energy electro-static fields. Unused devices must be stored in conductive foam or protective foam should be discharged to the destination socket before devices are removed. Devices should be handled at static safe workstations only. Unused digital inputs must be grounded or tied to the logic supply voltage. Unless otherwise noted, the voltage at any digital input should never exceed the supply voltage by more than 0.5 volts or go below -0.5 volts. If this condition cannot be maintained, limit input current on digital inputs by using series resistors or contact Hybrid Systems for technical assistance.

82C52
CMOS SERIAL
CONTROLLER INTERFACE



MARCH 1984

Advance Information

CMOS Serial Controller Interface

Features

- SINGLE CHIP UART/BRG
- DC TO 16MHz OPERATION
- CRYSTAL OR EXTERNAL CLOCK INPUT
- ON CHIP BAUD RATE GENERATOR . . . 72 SELECTABLE BAUD RATES
- INTERRUPT MODE WITH MASK CAPABILITY
- MICROPROCESSOR BUS ORIENTED INTERFACE
- 80C86 COMPATIBLE
- SCALED SAJI IV CMOS PROCESS
- SINGLE 5V POWER SUPPLY
- LOW POWER – 1mA/MHz TYPICAL
- MODEM INTERFACE
- LINE BREAK GENERATION AND DETECTION
- LOOPBACK AND ECHO MODES

Description

The 82C52 is a high performance programmable Universal Asynchronous Receiver/Transmitter (UART) and Baud Rate Generator (BRG) on a single chip. Utilizing the Harris advanced Scaled SAJI IV CMOS process, the 82C52 will support data rates from D.C. to 1M baud asynchronously with a 16X clock (0-16 MHz clock frequency).

The on-chip Baud Rate Generator can be programmed for any one of 72 different baud rates using a single industry standard crystal or external frequency source. A unique pre-scale divide circuit has been designed to provide standard RS-232-C baud rates when using any one of three industry standard baud rate crystals (1.8432 MHz, 2.4576 MHz, or 3.072 MHz).

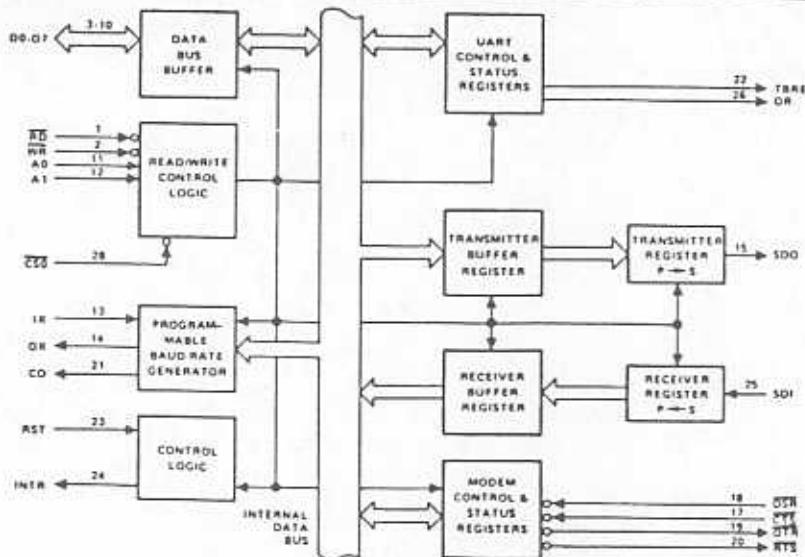
A programmable buffered clock output (CO) is available and can be programmed to provide either a buffered oscillator or 16X baud rate clock for general purpose system usage.

Inputs and outputs have been designed with full TTL/CMOS compatibility in order to facilitate mixed TTL/NMOS/CMOS system design.

Pinout

Top View	
RD	1
WR	2
DO	3
D1	4
D2	5
D3	6
D4	7
D5	8
D6	9
D7	10
AD	11
A1	12
TX	13
DX	14
CSO	28
VCC	27
DR	26
SDI	25
INTR	24
RST	23
TBRE	22
CO	21
RTS	20
DTR	19
DSR	18
CTS	17
GNU	16
SDO	15

Block Diagram



VCC - PIN 27
GND - PIN 16

CAUTION: These devices are sensitive to electrostatic discharge. Users should follow standard IC Handling Procedures.

82C52

82C52 PIN ASSIGNMENTS

PIN	I/O	SYMBOL	ACTIVE LEVEL	DESCRIPTION
1	I	\overline{RD}	Low	READ: The \overline{RD} input causes the 82C52 to output data to the data bus (D0-D7). The data output depends upon the state of the address inputs (A0, A1). CS0 enables the RD input.
2	I	\overline{WR}	Low	WRITE: The \overline{WR} input causes data from the data bus (D0-D7) to be input to the 82C52. Addressing and chip select action is the same as for read operations.
3-10	I/O	D0-D7	High	DATA BITS 0-7: The Data Bus provides eight, 3-state input/output lines for the transfer of data, control and status information between the 82C52 and the CPU. For character formats of less than 8 bits, the corresponding D7, D6 and D5 are considered "don't cares" for data WRITE operations and are 0 for data READ operations. These lines are normally in a high impedance state except during read operations. D0 is the Least Significant Bit (LSB) and is the first serial data bit to be received or transmitted.
11, 12	I	A0, A1	High	ADDRESS INPUTS: The address lines select the various internal registers during CPU bus operations.
13, 14	I, O	IX, OX		CRYSTAL/CLOCK: Crystal connections for the internal Baud Rate Generator. IX can also be used as an external clock input in which case OX should be left open.
15	O	SDO	High	SERIAL DATA OUTPUT: Serial data output from the 82C52 transmitter circuitry. A Mark (1) is a logic one (high) and Space (0) is a logic zero (low). SDO is held in the Mark condition when the transmitter is disabled when CTS is false, RST is true, when the Transmitter Register is empty, or when in the Loop Mode.
16		GND	Low	GROUND: Power supply ground connection.
17	I	\overline{CTS}	Low	CLEAR TO SEND: The logical state of the \overline{CTS} line is reflected in the CTS bit of the Modem Status Register. Any change of state in \overline{CTS} causes INTR to be set true when INTEN and MIEN are true. A false level on \overline{CTS} will inhibit transmission of data on the SDO output and will hold SDO in the Mark (high) state. If CTS goes false during transmission, the current character being transmitted will be completed. CTS does not affect Loop Mode operation.
18	I	\overline{DSR}	Low	DATA SET READY: The logical state of the \overline{DSR} line is reflected in the Modem Status Register. An change of state of DSR will cause INTR to be set if INTEN and MIEN are true. The state of this signal does not affect any other circuitry within the 82C52.
19	O	\overline{DTR}	Low	DATA TERMINAL READY: The \overline{DTR} signal can be set (low) by writing a logic 1 to the appropriate bit in the Modem Control Register (MCR). This signal is cleared (high) by writing a logic 0 to the DTR bit in the MCR or whenever a RST (high) is applied to the 82C52.
20	O	\overline{RTS}	Low	REQUEST TO SEND: The \overline{RTS} signal can be set (low) by writing a logic 1 to the appropriate bit in the MCR. This signal is cleared (high) by writing a logic 0 to the RTS bit in the MCR or whenever a reset (RST = high) is applied to the 82C52.
21	O	CO		CLOCK OUT: This output is user programmable to provide either a buffered IX output or a buffered Baud Rate Generator (16x) clock output. The buffered IX(Crystal or external clock source) output is provided when the Baud Rate Select Register (BRSR) bit 7 is set to a zero. Writing a logic one to BRSR bit 7 causes the CO output to provide a buffered version of the internal Baud Rate Generator clock which operates at sixteen times the programmed baud rate.
22	O	TBRE	High	TRANSMITTER BUFFER REGISTER EMPTY: The TBRE output is set (high) whenever the Transmitter Buffer Register (TBR) has transferred its data to the Transmit Register. Application of a reset (RST) to the 82C52 will also set the TBRE output. TBRE is cleared (low) whenever data is written to the TBR.
23	I	RST	High	RESET: The RST input forces the 82C52 into an "Idle" mode in which all serial data activities are suspended. The Modem Control Register (MCR) along with its associated outputs are cleared. The UART Status Register (USR) is cleared except for the TBRE and TC bits, which are set. The 82C52 remains in an "Idle" state until programmed to resume serial data activities. The RST input is a Schmitt trigger input.
24	O	INTR	High	INTERRUPT REQUEST: The INTR output is enabled by the INTEN bit in the Modem Control Register (MCR). The MIEN bit selectively enables modem status changes to provide an input to the INTR logic. Figure 2 shows the overall relationship of these interrupt control signals.
25	I	SDI	High	SERIAL DATA INPUT: Serial data input to the 82C52 receiver circuits. A Mark (1) is high, and a Space (0) is low. Data inputs on SDI are disabled when operating in the loop mode or when RST is true.

82C52

82C52 PIN ASSIGNMENTS

PIN	IO	SYMBOL	ACTIVE LEVEL	DESCRIPTION
26	O	DR	High	DATA READY: A true level indicates that a character has been received, transferred to the RBR and is ready for transfer to the CPU. DR is reset on a data READ of the Receiver Buffer Register (RBR) or when RST is true.
27		VCC	High	+ 5 VOLT SUPPLY: Positive power supply connection.
28	I	CS0	Low	CHIP SELECT: The chip select input acts as an enable signals for the RD and WR input signals.

RESET

During and after power-up, the 82C52 Reset Input (RST) should be held high for at least two IX clock cycles in order to initialize and drive the 82C52 circuits to an idle mode until proper programming can be done. A high on RST causes the following events to occur:

- Resets the internal Baud Rate Generator (BRG) circuits, clock counters and bit counters. The Baud Rate Select Register (BRSR) is not affected.
- Clears the UART Status Register (USR) except for Transmission Complete (TC) and Transmit Buffer Register Empty (TBRE) which are set. The Modem Control Register (MCR) is also cleared. All of the discrete lines, memory elements and miscellaneous logic associated with these register bits are also cleared or turned off. Note that the UART Control Register (UCR) is not affected.

Following removal of the reset condition (RST = low), the 82C52 remains in the idle mode until programmed to its desired system configuration.

PROGRAMMING THE 82C52

The complete functional definition of the 82C52 is programmed by the systems software. A set of control words (UCR, BRSR and MCR) must be sent out by the CPU to initialize the 82C52 to support the desired communication format. These control words will program the character length, number of stop bits, even/odd/no parity, baud rate etc. Once programmed, the 82C52 is ready to perform its communication functions.

The control registers can be written to in any order. However, the MCR should be written to last because it controls the interrupt enables, modem control outputs and the receiver enable bit. Once the 82C52 is programmed and operational, these registers can be updated any time the 82C52 is not immediately transmitting or receiving data.

Table 1 shows the control signals required to access 82C52 internal registers.

CS0	A1	A0	WR	RD	OPERATION
0	0	0	0	1	Data Bus → Transmitter Buffer Register (TBR)
0	0	0	1	0	Receiver Buffer Register (RBR) → Data Bus
0	0	1	0	1	Data Bus → UART Control Register (UCR)
0	0	1	1	0	UART Status Register (USR) → Data Bus
0	1	0	0	1	Data Bus → Modem Control Register (MCR)
0	1	0	1	0	MCR → Data Bus
0	1	1	0	1	Data Bus → Bit Rate Select Register (BRSR)
0	1	1	1	0	Modem Status Register (MSR) → Data Bus

TABLE 1

UART CONTROL REGISTER (UCR)

The UCR is a write only register which configures the UART transmitter and receiver circuits. Data bits D7 and D6 are not used but should always be set to a logic zero (0) in order to insure software compatibility with future product upgrades. During the Echo Mode, the transmitter always repeats the received word and parity, even when the UCR is programmed with different or no parity.

UCR

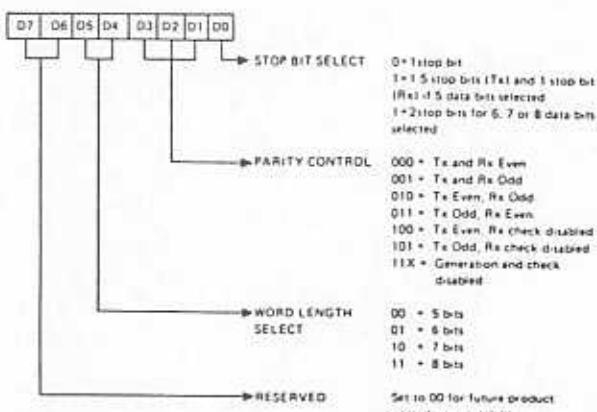


FIGURE 1

BAUD RATE SELECT REGISTER (BRSR)

The 82C52 is designed to operate with a single crystal or external clock driving the IX input pin. The Baud Rate Select Register is used to select the divide ratio (one of 72) for the internal Baud Rate Generator circuitry. The internal circuitry is separated into two separate counters, a Prescaler and a Divisor Select. The Prescaler can be set to any one of four division rates, $\div 1$, $\div 3$, $\div 4$ or $\div 5$.

The Prescaler design has been optimized to provide standard baud rates using any one of three popular crystal frequencies. By using one of these common system clock frequencies, 1.8432 MHz, 2.4576 MHz or 3.072 MHz and Prescaler divide ratios of $\div 3$, $\div 4$, or $\div 5$ respectively, the Prescaler output will provide a constant 614.4 KHz. When this frequency is further divided by the Divisor Select counter, any of the standard baud rates from 50 Baud to 38.4 KBaud can be selected (see Table 2). Non-standard baud rates up to 1 Mbaud can be selected by using different input frequencies (crystal or an external frequency input up to 16 MHz) and/or different Prescaler and Divisor Select ratios.

Regardless of the baud rate, the baud rate generator provides a clock which is 16 times the desired baud rate. For example, in order to operate at a 1 Mbaud data rate, a 16 MHz crystal, a Prescale rate of $\div 1$, and a Divisor Select rate of "external" would be used. This would provide a 16 MHz clock as the output of the Baud Rate Generator to the Transmitter and Receiver circuits.

The CO select bit in the BRSR selects whether a buffered version of the external frequency input (IXinput) or the Baud Rate Generator output (16x baud rate clock) will be output on the CO output (pin 21). The Baud Rate Generator output will always be a 50% nominal duty cycle except when "external" is selected and the Prescaler is set to $\div 3$ or $\div 5$.

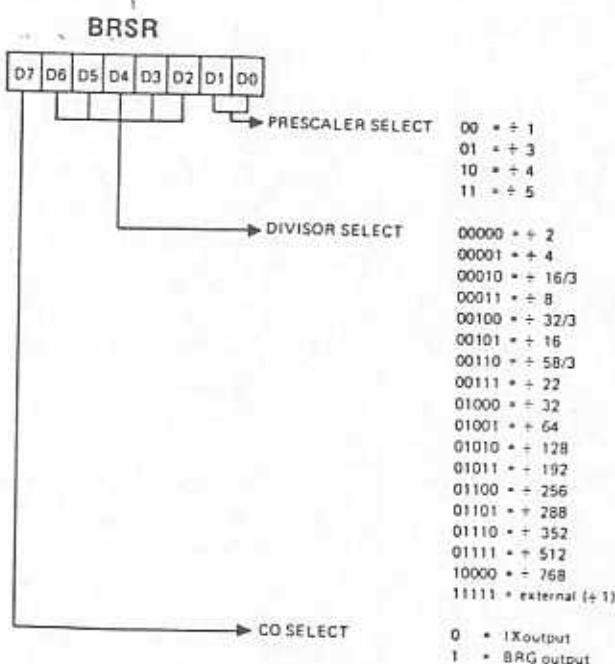


FIGURE 2

BAUD RATE	DIVISOR
38.4K	external
19.2K	2
9600	4
7200	16/3
4800	8
3600	32/3
2400	16
2000*	58/3
1800	22
1200	32
600	64
300	128
200	192
150	256
134.5*	288
110*	352
75	512
50	768

TABLE 2

Note: These baud rates are based upon the following input frequency/Prescale divisor combinations.
 1.8432 MHz and Prescale = $\div 3$
 2.4576 MHz and Prescale = $\div 4$
 3.072 MHz and Prescale = $\div 5$

*All baud rates are exact except for:

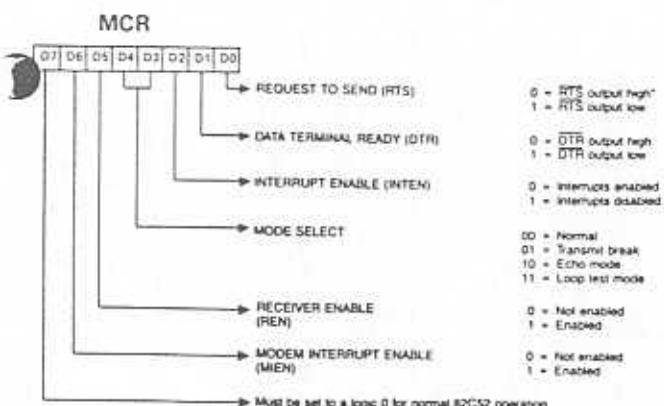
BAUD RATE	ACTUAL	PERCENT ERROR
2000	1986.2	0.69%
134.5	133.33	0.87%
110	109.71	0.26%

MODEM CONTROL REGISTER

The MCR is a general purpose control register which can be written to and read from. The RTS and DTR outputs are directly controlled by their associated bits in this register. Note that a logic one asserts a true logic level (low) at these output pins. The Interrupt Enable (INTEN) bit is the overall control for the INTR output pin. When INTEN is false, INTR is held false (low).

The Operating Mode bits configure the 82C52 into one of four possible modes. "Normal" configures the 82C52 for normal full or half duplex communications. "Transmit Break" enables the transmitter to only transmit break characters (Start, Data and Stop bits all are logic zero). The Echo Mode causes any data that is received on the SDI input pin to be re-transmitted on the SDO output pin. Note that this output is a buffered version of the data seen on the SDI input and is not a resynchronized output. Also note that normal UART transmission via the Transmitter Register is disabled when operating in the Echo mode (see Figure 4). The Loop Test Mode internally routes transmitted data to the receiver circuitry for the purpose of self test. The transmit data is disabled from the SDO output pin. The Receiver Enable bit gates off the input to the receiver circuitry when in the false state.

Modem Interrupt Enable will permit any change in modem status line inputs (CTS, DSR) to cause an interrupt when this bit is enabled. Bit D7 must always be written to with a logic zero to insure correct 82C52 operation.



*See Modem Status Register description (p5) for a description of register flag images with respect to output pins.

FIGURE 3

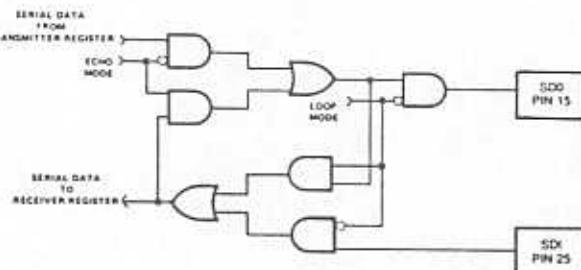


FIGURE 4 – Loop and Echo Mode Functionality

UART STATUS REGISTER (USR)

The USR provides a single register that the controlling system can examine to determine if errors have occurred or if other status changes in the 82C52 require attention. For this reason, the USR is usually the first register read by the CPU to determine the cause of an interrupt or to poll the status of the 82C52.

Three error flags OE, FE and PE report the status of any error conditions detected in the receiver circuitry. These error flags are updated with every character received during reception of the stop bits. The Overrun Error (OE) indicates that a character in the Receiver Register has been received and cannot be transferred to the Receiver Buffer Register (RBR) because the RBR was not read by the CPU. Framing Error (FE) indicates that the last character received contained improper stop bits. This could be caused by the absence of the required stop bit(s) or by a stop bit(s) that was too short to be properly detected. Parity Error (PE) indicates that the last character received contained a parity error based on the programmed parity of the receiver and the calculated parity of the received character data and parity bits.

The Received Break (RBRK) status bit indicates that the last character received was a break character. A break character would be considered to be an invalid data character in that the entire character including parity and stop bits are a logic zero.

The Modem Status bit is set whenever a transition is detected on any of the Modem input lines (CTS or DSR). A subsequent read of the Modem Status Register will show the state of these two signals. Assertion of this bit will cause an interrupt (INTR) to be generated if the MIEN and INTEN bits in the MCR register are enabled.

The Transmission Complete (TC) bit indicates that both the TBR and Transmitter Registers are empty and the 82C52 has completed transmission of the last character it was commanded to transmit. The assertion of this bit will cause an interrupt (INTR) if the INTEN bit in the MCR register is true.

The Transmitter Buffer Register Empty (TBRE) bit indicates that the TBR register is empty and ready to receive another character.

The Data Ready (DR) bit indicates that the RBR has been loaded with a received character (including Break) and that the CPU may access this data.

Assertion of the TBRE or DR bits do not affect the INTR logic and associated INTR output pin since the 82C52 has been designed to provide separate requests via the DR and TBRE output pins. If a single interrupt for any status change in the 82C52 is desired this can be accomplished by "ORing" DR, TBRE and INTR together.

Reading the USR clears all of the status bits in the USR register but does not affect associated output pins.

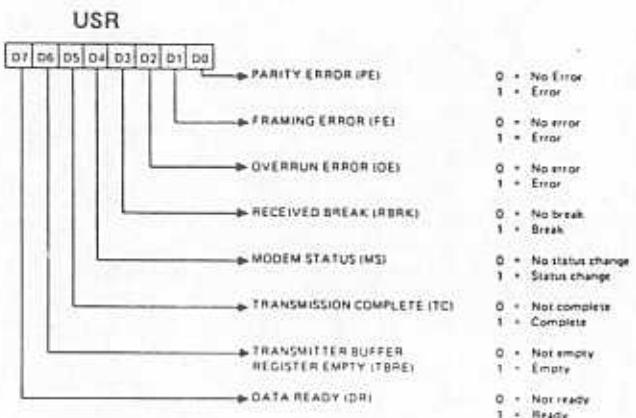


FIGURE 5

MODEM STATUS REGISTER (MSR)

The MSR allows the CPU to read the modem signal inputs by accessing the data bus interface of the 82C52. Like all of the register images of external pins in the 82C52, true logic levels are represented by a high (1) signal level. By following this consistent definition, the system software need not be concerned with whether external signals are high or low true. In particular, the modem signal inputs are low true, thus a 0 (true assertion) at a modem input pin is represented by a 1 (true) in the MSR.

Any change of state in any modem input signals will set the Modem Status (MS) bit in the USR register. When this happens, an interrupt (INTR) will be generated if the MIEN and INTEN bits of the MCR are enabled.

The Data Set Ready (DSR) input is a status indicator from the modem to the 82C52 which indicates that the modem is ready to provide received data to the 82C52 receiver circuitry.

Clear to Send (CTS) is both a status and control signal from the modem that tells the 82C52 that the modem is ready to receive transmit data from the 82C52 transmitter output (SDO). A high (false) level on this input will inhibit the 82C52 from beginning transmission and if asserted in the middle of a transmission will only permit the 82C52 to finish transmission of the current character.

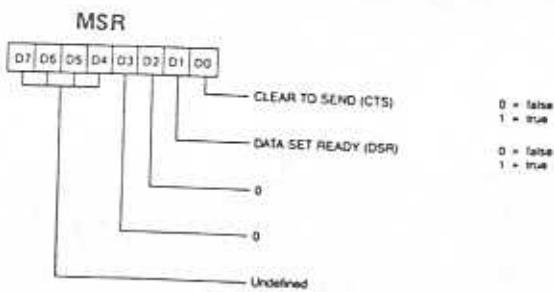


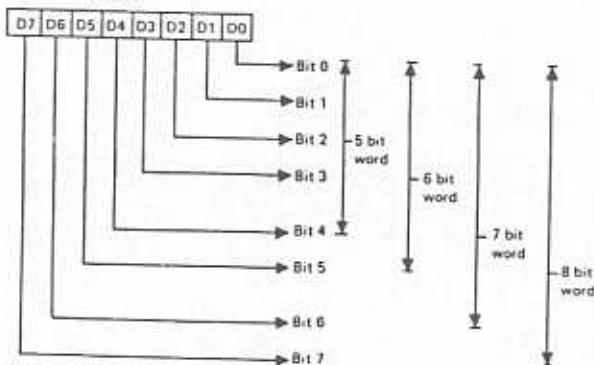
FIGURE 6

RECEIVER BUFFER REGISTER (RBR)

The receiver circuitry in the 82C52 is programmable for 5, 6, 7 or 8 data bits per character. For words of less than 8 bits, the data is right justified to the Least Significant Bit (LSB = D0). Bit D0 of a data word is always the first data bit received. The unused bits in a less than 8 bit word, at the parallel interface, are set to a logic zero (0) by the 82C52.

Received data at the SDI input pin is shifted into the Receiver Register by an internal 1x clock which has been synchronized to the incoming data based on the position of the start bit. When a complete character has been shifted into the Receiver Register, the assembled data bits are parallel loaded into the Receiver Buffer Register. Both the DR output pin and DR flag in the USR register are set. This double buffering of the received data permits continuous reception of data without losing any of the received data.

While the Receiver Register is shifting a new character into the 82C52, the Receiver Buffer Register is holding a previously received character for the system CPU to read. Failure to read the data in the RBR before complete reception of the next character can result in the loss of the data in the Receiver Register. The OE flag in the USR register indicates the overrun condition.

RBR

Note: The LSB, Bit 0 is the first serial data bit received.

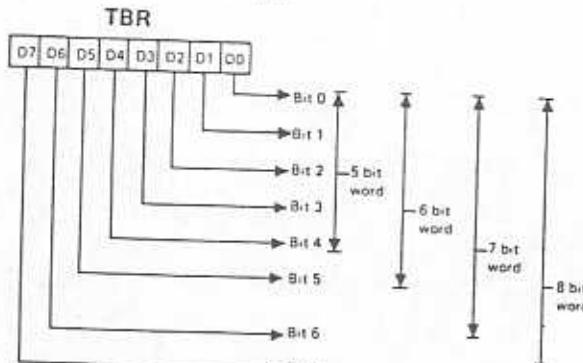
FIGURE 7

TRANSMITTER BUFFER REGISTER (TBR)

The Transmitter Buffer Register (TBR) accepts parallel data from the data bus (D0-D7) and holds it until the Transmitter Register is empty and ready to accept a new character for transmission. The transmitter always has the same word length and number of stop bits as the receiver. For words of less than 8 bits the unused bits at the microprocessor data bus are ignored by the transmitter.

Bit 0, which corresponds to D0 at the data bus, is always the first serial data bit transmitted. Provision is made for the transmitter parity to be the same or different from the receiver. The TBRE output pin and flag (USR register) reflect the status

of the TBR. The TC flag (USR register) indicates when both the TBR and TR are empty.



Note: The LSB, Bit 0 is the first serial data bit transmitted.

FIGURE 8

82C52 INTERRUPT STRUCTURE

The 82C52 has provisions for software masking of interrupts generated for the INTR output pin. Two control bits in the MCR register, MIEN and INTEN, control modem status interrupts and overall 82C52 interrupts respectively. Figure 9 illustrates the logical control function provided by these signals.

The modem status inputs (DSR and CTS) will trigger the edge detection circuitry with any change of status. Reading the MSR register will clear the detect circuit but has no effect on the status bits themselves. These status bits always reflect the state of the input pins regardless of the mask control signals. Note that the state (high or low) of the status bits are inverted versions of the actual input pins.

The edge detection circuits for the USR register signals will trigger only for a positive edge (true assertion) of these status bits. Reading the USR register not only clears the edge detect circuit but also clears (sets to 0) all of the status bits. The output pins associated with these status bits are not affected by reading the USR register.

A hardware reset of the 82C52 sets the TC status bit in the USR. When interrupts are subsequently enabled an interrupt can occur due to the fact that the positive edge detection circuitry in the interrupt logic has detected the setting of the TC bit. If this interrupt is not desired the USR should be read prior to enabling interrupts. This action resets the positive edge detection circuitry in the interrupt control logic (Figure 9).

NOTE: For USR and MSR, the setting of status bits is inhibited during status register READ operations. If a status condition is generated during a READ operation, the status bit is not set until the trailing edge of the RD pulse.

If the bit was already set at the time of the READ operation, and the same status condition occurs, that status bit will be cleared at the trailing edge of the RD pulse instead of being set again.

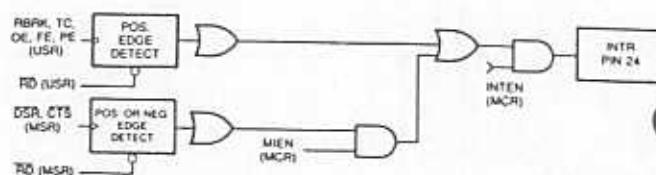


FIGURE 9 – 82C52 Interrupt Structure

SOFTWARE RESET

A software reset of the 82C52 is a useful method for returning to a completely known state without exercising a complete system reset. Such a reset would consist of writing to the UCR, BRSR and MSR registers. The USR and RBR registers should be read prior to enabling interrupts in order to clear out any residual data or status bits which may be invalid for subsequent operation.

CRYSTAL OPERATION

The 82C52 crystal oscillator circuitry is designed to operate with a fundamental mode, parallel resonant crystal. This circuit is the same one used in the Harris 82C84A clock generator/driver and the general oscillator operation information which is contained in Tech Brief TB-47 will be pertinent to the 82C52. To summarize, Table 3 and Figure 10 show the required crystal parameters and crystal circuit configuration respectively.

When using an external clock source, the IX input is driven and the OX output is left open. Power consumption when using an external clock is typically 50% of that required when using a crystal. This is due to the sinusoidal nature of the drive circuitry when using a crystal.

82C52 - 80C86 INTERFACING

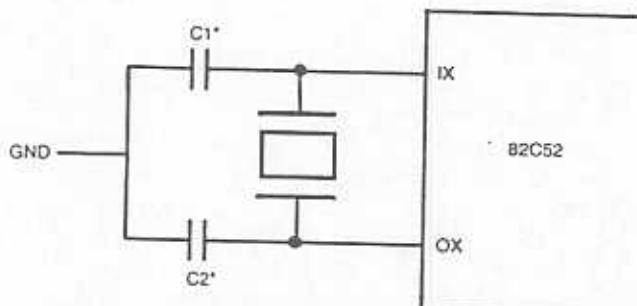
The following example (Figure 11) shows the interface for an 82C52 in an 80C86 system.

Use of the Harris CMOS Interrupt Controller (82C59A) is optional and necessary only if an interrupt driven system is desired.

By using the Harris CMOS 82C84A clock generator, the system can be built with a single crystal providing both the processor clock and the clock for the 82C52. The 82C52 has

PARAMETER	TYPICAL CRYSTAL SPECIFICATION
Frequency	1.0 to 16MHz
Type of Operation	Parallel resonant, Fundamental mode
Load Capacitance (CL)	20 or 32 pF (typ.)
Rseries (Max.)	100 ohms (f=16 MHz, CL = 32pF)
	200 ohms (f=16 MHz, CL = 20pF)

TABLE 3



*C1 = C2 = 20 pF for CL = 20 pF

*C1 = C2 = .47 pF for CL = 32 pF

FIGURE 10

special divider circuitry which is designed to supply industry standard baud rates with a 2.4576 MHz input frequency. Using a 15 MHz crystal as shown, results in less than a 2% frequency error which is adequate for many applications. For more precise baud rate requirements, a 14.7456 MHz crystal will drive the 80C86 at 4.9 MHz and provide the 82C52 with the standard baud rate input frequency of 2.4576 MHz. If baud rates above 156 Kbaud are desired, the OSC output can be used instead of the PCLK ($\div 6$) output for asynchronous baud rates up to 1 Mbaud.

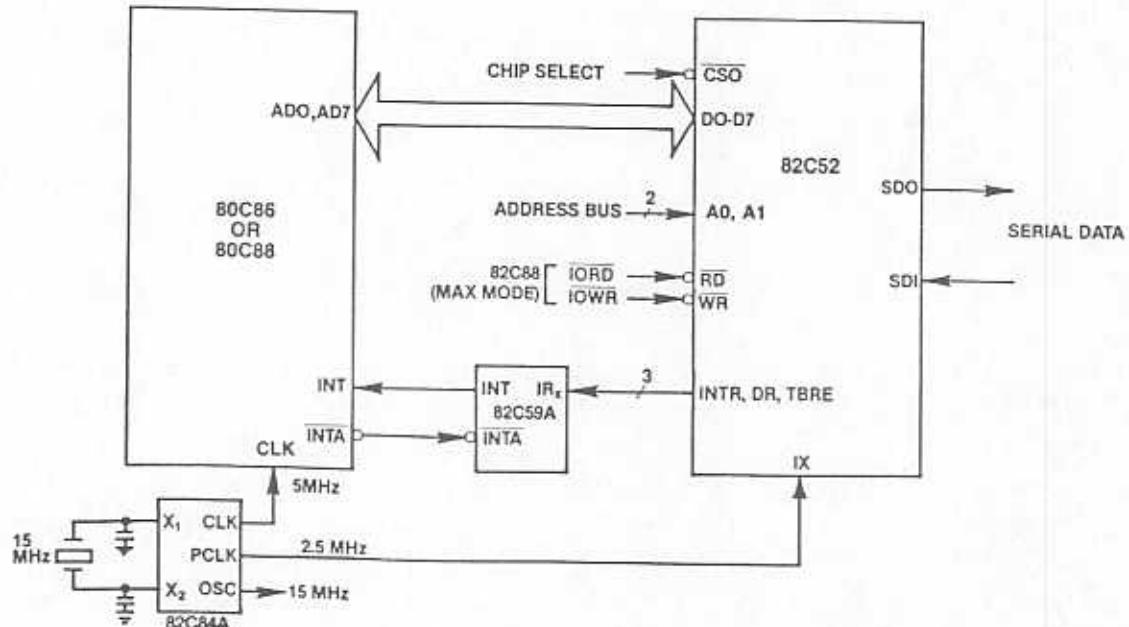


FIGURE 11 - 80C86/82C52 Interface

Specifications

ABSOLUTE MAXIMUM RATINGS

Supply Voltage	+8.0 VOLTS	Operating Temperature Range	
Operating Voltage Range	+4V to +7V	Commercial	0°C to +70°C
Input Voltage Applied	GND - 2.0V to 6.5V	Industrial	-40°C to +85°C
Output Voltage Applied	GND - 0.5V to VCC+0.5V	Military	-55°C to +125°C
Storage Temperature Range	-65°C to +150°C	Maximum Power Dissipation	1 Watt

CAUTION: Stresses above those listed in the "ABSOLUTE MAXIMUM RATINGS" may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

D.C. ELECTRICAL CHARACTERISTICS

VCC = 5.0V ± 10%; TA = 0°C to +70°C (C82C52); TA = -40°C to +85°C (I82C52); TA = -55°C to +125°C (M82C52)

SYMBOL	PARAMETER	MIN	MAX	UNITS	TEST CONDITIONS
VIH	LOGICAL ONE INPUT VOLTAGE	2.0 2.2		V	I82C52, C82C52 M82C52
VIL	LOGICAL ZERO INPUT VOLTAGE		0.8	V	
VTH	SCHMITT TRIGGER LOGICAL ONE INPUT VOLTAGE	VCC - 0.5		V	RESET INPUT
VTL	SCHMITT TRIGGER LOGICAL ZERO INPUT VOLTAGE		GND + 0.5	V	RESET INPUT
VIH (CLK)	LOGICAL ONE CLOCK INPUT VOLTAGE	VCC - 0.5		V	EXTERNAL CLOCK
VIL (CLK)	LOGICAL ZERO CLOCK INPUT VOLTAGE		GND + 0.5	V	EXTERNAL CLOCK
VOH	OUTPUT HIGH VOLTAGE	3.0 VCC - 0.4		V	IOH = -2.5 mA IOH = -100 µA
VOL	OUTPUT LOW VOLTAGE		0.4	V	IOL = +2.5 mA
IIL	INPUT LEAKAGE CURRENT	-1.0	+1.0	µA	OV ≤ VIN ≤ VCC
IOL	OUTPUT LEAKAGE CURRENT	-10.0	+10.0	µA	OV ≤ VO ≤ VCC
ICCOP*	OPERATING POWER SUPPLY CURRENT		3	mA	EXTERNAL CLOCK F = 2.4575 MHz VCC = 5.5V VIN = VCC or GND OUTPUTS OPEN

*Guaranteed and sampled, but not 100% tested. ICCOP is typically <1 mA/MHz.

CAPACITANCE

TA = 25°C; VCC = GND = OV; VIN = +5V or GND

SYMBOL	PARAMETER	MIN	MAX	UNITS	TEST CONDITIONS
CIN*	INPUT CAPACITANCE		10	pf	FREQ = 1 MHz Unmeasured pins returned to GND
Cout*, Cvo*	OUTPUT CAPACITANCE I/O CAPACITANCE		15 20	pf pf	

*Guaranteed and sampled, but not 100% tested

A.C. CHARACTERISTICS

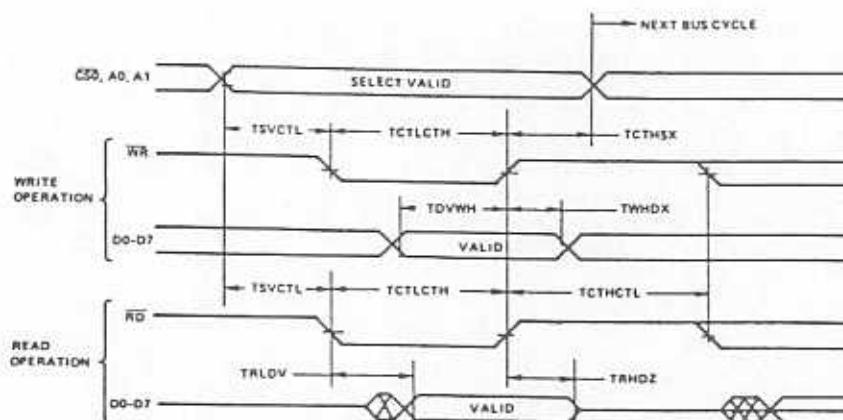
V_{CC} = 5.0V ± 10%; T_A = 0°C to +70°C (C82C52); T_A = -40°C to +85°C (I82C52);
 T_A = -55°C to +125°C (M82C52)

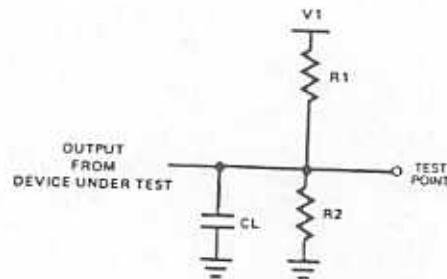
Timing Requirements

SYMBOL	PARAMETER	MIN	MAX	UNITS	TEST CONDITIONS
TSVCTL	Select Setup to Control Leading Edge	30		ns	
TCTHSX	Select Hold From Control Trailing Edge	50		ns	
TCTLCTH	Control Pulse Width	150		ns	Control Consists of RD or WR
TCTHCTL	Control Disable to Control Enable	100		ns	
TRLDV	Read Low to Data Valid		120	ns	1
TRHDZ	Read Disable	0	60	ns	2
TDVWH	Data Setup Time	50		ns	
TWHDX	Data Hold Time	20		ns	
FC	Clock Frequency	0	16	MHz	TCHCL + TCLCH must be ≥ 62.5 ns
TCHCL	Clock High Time	25		ns	
TCLCH	Clock Low Time	25		ns	
TR/TF	IX Input Rise/Fall Time (External Clock)		tx	ns	tx ≤ $\frac{1}{6FC}$ or 50ns whichever is smaller
TFCO	Clock Output Fall Time		15	ns	CL = 50 pF
TRCO	Clock Output Rise Time		15	ns	CL = 50 pF

Timing Diagram

BUS OPERATION

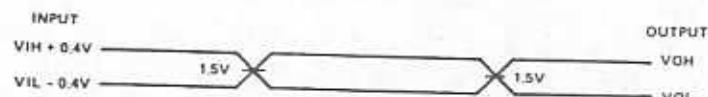


A.C. Test Circuits

TEST CONDITION	V1	R1	R2	CL
1 Propagation Delay	1.7V	520	∞	100pF
2 Disable Delay	VCC	5K	5K	50pF

A.C. Testing Input, Output Waveform

PROPAGATION DELAY



ENABLE/DISABLE DELAY

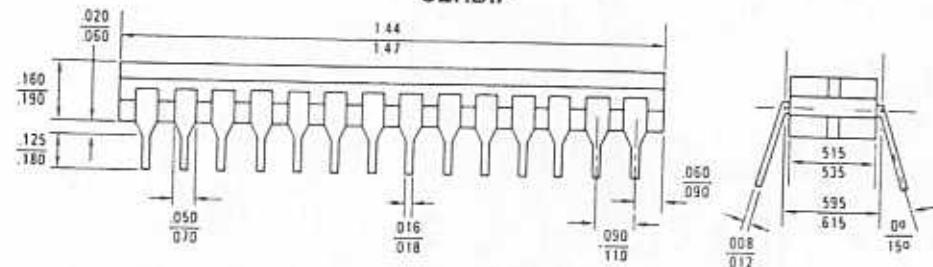


A.C. Testing: All input signals must switch between VIL - 0.4V and VIH + 0.4V. TR and TF must be less than or equal to 15ns.

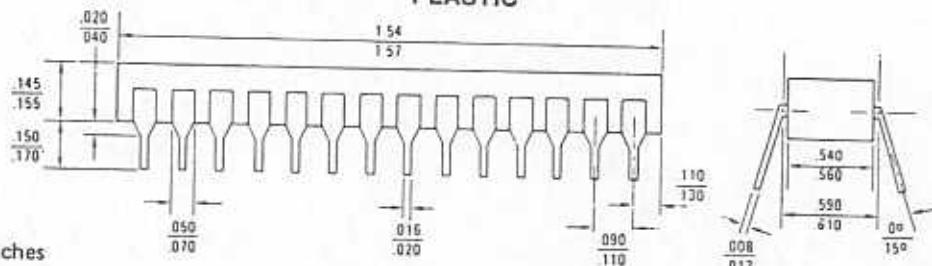
28 LEAD CERDIP/PLASTIC
82C52

Package

CERDIP



PLASTIC



NOTE: MIN.
MAX.

All dimensions in inches

Ordering Information

M	D	82C52	/B
TEMPERATURE RANGE	PACKAGE TYPE	PART NUMBER	DASH 8 PROGRAM-
C - Commercial	P - Plastic	82C52	
I - Industrial	D - Ceramic		
M - Military	X - Die Form		
X - 25°C	R - Leadless Chip Carrier		

NOTICE: Harris Semiconductor's products are sold by description only. Harris Semiconductor reserves the right to make changes in circuit design and/or specifications at any time without notice. Accordingly, the reader is cautioned to verify that data sheets are current before placing orders.

Notes

Notes

SALES OFFICES

SUITE 5
8090 NORTH 85TH WAY
PHOENIX, AZ 85258
(602) 998-9766

SUITE 320
1503 SOUTH COAST DRIVE
COSTA MESA, CA 92626
(714) 540-2176

SUITE B120
883 STIERLIN ROAD
MOUNTAIN VIEW, CA 94043
(415) 964-6443

SUITE 205
6400 CANDOGA AVENUE
WOODLAND HILLS, CA 92367
(213) 992-0686

SUITE 215
3890 WEST COMMERCIAL BLVD.
FORT LAUDERDALE, FL 33309
(305) 739-0016

SUITE 400
875 JOHNSON FERRY ROAD
ATLANTA, GA 30342
(404) 256-4000

SUITE 300
6400 SHAFER COURT
ROSEMOUNT, IL 60018
(312) 692-4960

SUITE 101
1717 EAST 116TH STREET
CARMEL, IN 46032
(317) 844-8011

SUITE 308
1 BURLINGTON WOODS DRIVE
BURLINGTON, MA 01803
(617) 273-5942

SUITE 703
2850 METRO DRIVE
MINNEAPOLIS, MN 55420
(612) 854-3558

SUITE 273
555 BROADHOLLOW ROAD
MELVILLE, L.I., NY 11747
(516) 249-4500

SUITE 1101
996 OLD EAGLE SCHOOL ROAD
WAYNE, PA 19087
(215) 687-6680

SUITE 110
17120 DALLAS PARKWAY
DALLAS, TX 75248
(214) 248-3237

33919 9TH AVENUE SOUTH
FEDERAL WAY, WA 98003
(206) 838-4878

SUITE 8
2005 BROADWAY
VANCOUVER, WA 98663
(206) 696-0043

NOTICE:

Harris Semiconductor's products are sold by description only. Harris Semiconductor reserves the right to make changes in circuit design and/or specifications at any time without notice. Accordingly, the reader is cautioned to verify that data sheets are current before placing orders.

For maintenance of performance and reliability, Harris Semiconductor strongly recommends that the "I.C. Handling Procedures", located in Section I of the current Analog Products Data Book, be followed closely by any activity involved with I.C. products.

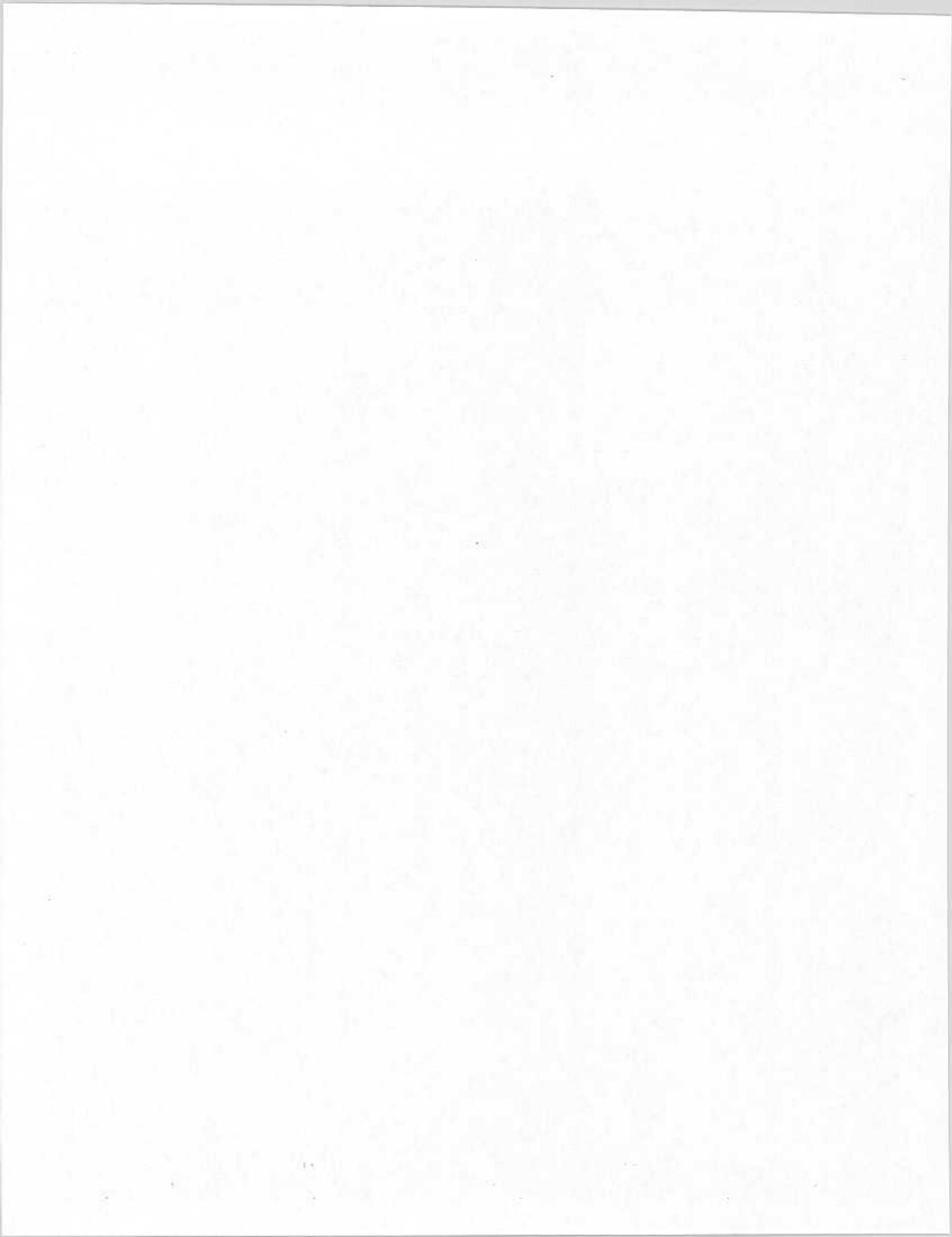


HARRIS

SEMICONDUCTOR DIGITAL PRODUCTS DIVISION

82C54

**PROGRAMMABLE
INTERVAL TIMER**



**CMOS PROGRAMMABLE
INTERVAL TIMER**

September 1985

Features

- COMPATIBLE WITH NMOS 8254
 - ENHANCED VERSION OF NMOS 8253
- THREE INDEPENDENT 16 BIT COUNTERS
- SIX PROGRAMMABLE COUNTER MODES
- STATUS READ BACK COMMAND
- BINARY OR BCD COUNTING
- FULLY TTL COMPATIBLE
- SCALED SAJI IV CMOS PROCESS
- LOW POWER
 - ICCSB = $10\mu A$
 - ICCOP = $10mA$
- SINGLE 5V POWER SUPPLY
- COMMERCIAL, INDUSTRIAL AND MILITARY TEMPERATURE RANGES

Pinout

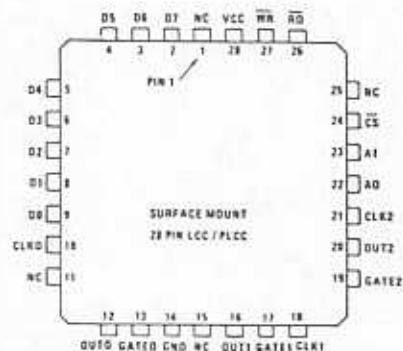
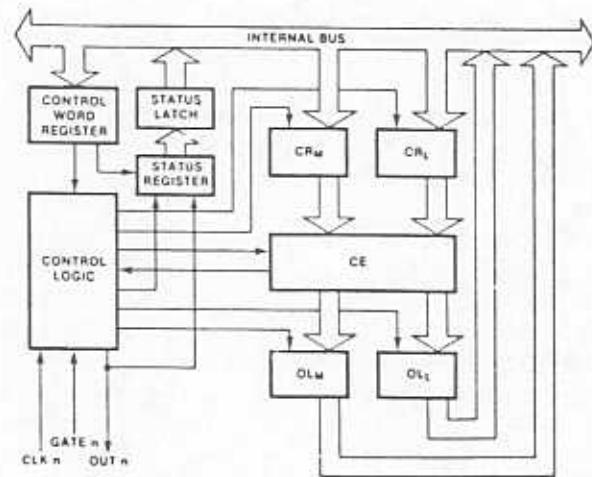
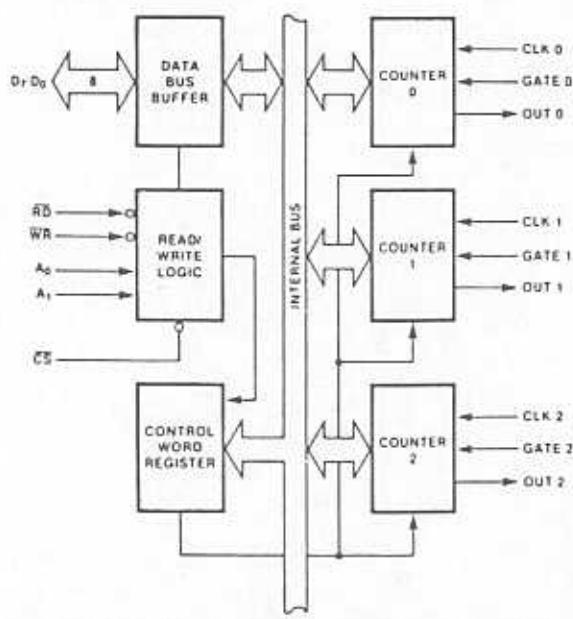
07	1	24	VCC
06	2	23	WR
05	3	22	RD
04	4	21	CS
03	5	20	A1
02	6	19	A0
01	7	18	CLK 2
00	8	17	OUT 2
CLK 0	9	16	GATE 2
OUT 0	10	15	CLK 1
GATE 0	11	14	GATE 1
GND	12	13	OUT 1

Description

The Harris 82C54 is a high performance CMOS Programmable Interval Timer manufactured using a self-aligned silicon gate CMOS process (Scaled SAJI IV). The 82C54 has three independently programmable and functional 16 bit counters, each capable of handling clock input frequencies of up to 8MHz. The high speed and industry standard configuration of the 82C54 make it compatible with the Harris 80C86 and 80C88 CMOS microprocessors along with many other industry standard processors.

Six programmable timer modes allow the 82C54 to be used as an event counter, elapsed time indicator, programmable one-shot along with many other applications.

Static CMOS circuit design insures low operation power. Harris advanced SAJI process results in a significant reduction in power with performance equal to or greater than existing equivalent products.


Functional Diagram


COUNTER INTERNAL BLOCK DIAGRAM

Pin Description

SYMBOL	PIN NUMBER	TYPE	NAME AND FUNCTION
D7-D0	1-8	I/O	Data: Bi-directional three state data bus lines, connected to system data bus.
CLK0	9	I	Clock 0: Clock input of Counter 0.
OUT 0	10	O	Output 0: Output of Counter 0.
GATE 0	11	I	Gate 0: Gate input of Counter 0.
GND	12		Ground: Power supply connection.
OUT 1	13	O	Out 1: Output of Counter 1.
GATE 1	14	I	Gate 1: Gate input of Counter 1.
CLK 1	15	I	Clock 1: Clock input of Counter 1.
GATE 2	16	I	Gate 2: Gate input of Counter 2.
OUT 2	17	O	Out 2: Output of Counter 2.
CLK 2	18	I	Clock 2: Clock input of Counter 2.
A0, A1	19-20	I	Address: Select inputs for one of the three counters or Control Word Register for read/write operations. Normally connected to the system address bus. <u>A1 A0 Selects</u> 0 0 Counter 0 0 1 Counter 1 1 0 Counter 2 1 1 Control Word Register
CS	21	I	Chip Select: A low on this input enables the 82C54 to respond to RD and WR signals. RD and WR are ignored otherwise.
RD	22	I	Read: This input is low during CPU read operations.
WR	23	I	Write: This input is low during CPU write operations.
VCC	24		Power: +5V power supply connection.

*Functional Description***General**

The 82C54 is a programmable interval timer/counter designed for use with microcomputer systems. It is a general purpose, multi-timing element that can be treated as an array of I/O ports in the system software.

The 82C54 solves one of the most common problems in any microcomputer system, the generation of accurate time delays under software control. Instead of setting up timing loops in software, the programmer configures the 82C54 to match his requirements and programs one of the counters for the desired delay. After the desired delay, the 82C54 will interrupt the CPU. Software overhead is minimal and variable length delays can easily be accommodated.

Some of the other computer/timer functions common to

microcomputers which can be implemented with the 82C54 are:

- Real time clock
- Event counter
- Digital one-shot
- Programmable rate generator
- Square wave generator
- Binary rate multiplier
- Complex waveform generator
- Complex motor controller

Data Bus Buffer

This 3-state, bi-directional, 8-bit buffer is used to interface the 82C54 to the system bus (see Figure 1).

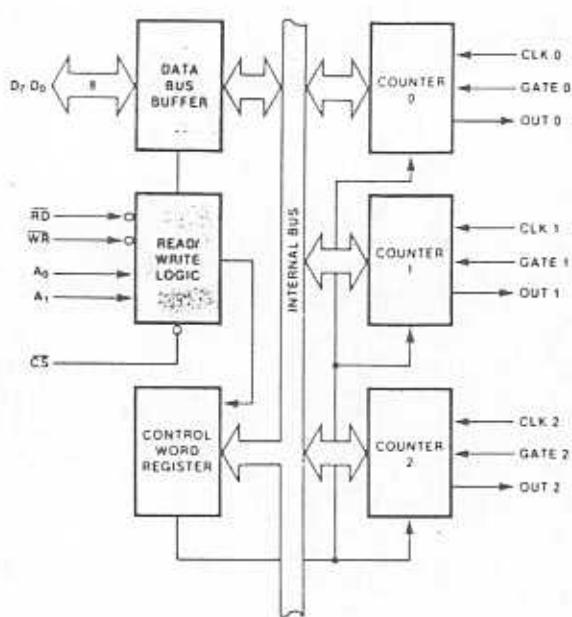


Figure 1. Data Bus Buffer and Read/Write Logic Function

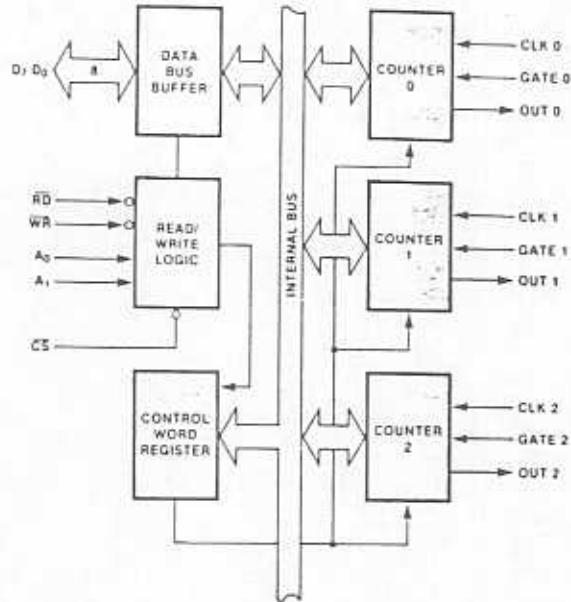


Figure 2. Control Word Register and Counter Functions

Read/Write Logic

The Read/Write Logic accepts inputs from the system bus and generates control signals for the other functional blocks of the 82C54. A1 and A0 select one of the three counters or the Control Word Register to be read from/written into. A "low" on the RD input tells the 82C54 that the CPU is reading one of the counters. A "low" on the WR input tells the 82C54 that the CPU is writing either a Control Word or an initial count. Both RD and WR are qualified by CS; RD and WR are ignored unless the 82C54 has been selected by holding CS low.

Control Word Register

The Control Word Register (Figure 2) is selected by the Read/Write Logic when A1, A0 = 11. If the CPU then does a write operation to the 82C54, the data is stored in the Control Word Register and is interpreted as a Control Word used to define the Counter operation.

The Control Word Register can only be written to; status information is available with the Read-Back Command.

Counter 0, Counter 1, Counter 2

These three functional blocks are identical in operation, so only a single Counter will be described. The internal block diagram of a single counter is shown in Figure 3.

The Counters are fully independent. Each Counter may operate in a different Mode.

The Control Word Register is shown in the figure; it is not part of the Counter itself, but its contents determine how the Counter operates.

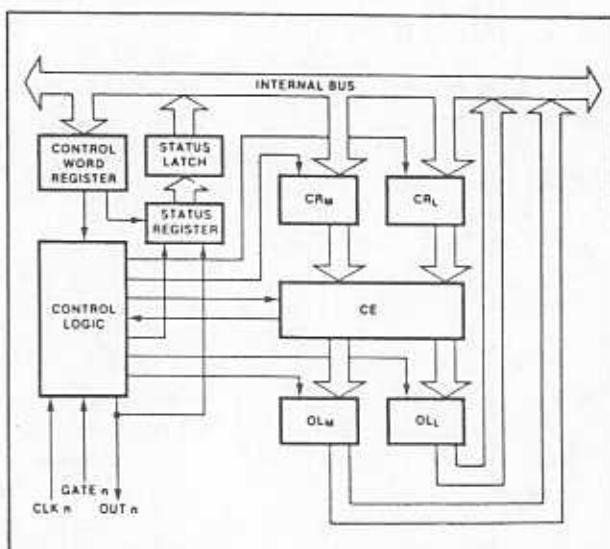


Figure 3. Counter Internal Block Diagram

The status register, shown in the Figure, when latched, contains the current contents of the Control Word Register and status of the output and null count flag. (See detailed explanation of the Read-Back command.)

The actual counter is labeled CE (for "Counting Element). It is a 16-bit presettable synchronous down counter.

OLM and OLL are two 8-bit latches. OL stands for "Output Latch"; the subscripts M and L for "Most significant byte" and "Least significant byte" respectively. Both are normally referred to as one unit and called just OL. These latches normally "follow" the CE, but if a suitable Counter Latch Command is sent to the 82C54, the latches "latch" the present count until read by the CPU and then return to "following" the CE. One latch at a time is enabled by the counter's Control Logic to drive the internal bus. This is how the 16-bit Counter communicated over the 8-bit internal bus. Note that the CE itself cannot be read; whenever you read the count, it is the OL that is being read.

Similarly, there are two 8-bit registers called CRM and CRL (for "Count Register"). Both are normally referred to as one unit and called just CR. When a new count is written to the Counter, the count is stored in the CR and later transferred to the CE. The Control Logic allows one register at a time to be loaded from the internal bus. Both bytes are transferred to the CE simultaneously. CRM and CRL are cleared when the Counter is programmed for one byte counts (either most significant byte only or least significant byte only) the other byte will be zero. Note that the CE cannot be written into; whenever a count is written, it is written into the CR.

The Control Logic is also shown in the diagram. CLK n, GATE n, and OUT n are all connected to the outside world through the Control Logic.

82C54 System Interface

The 82C54 is treated by the system software as an array of peripheral I/O ports; three are counters and the fourth is a control register for MODE programming.

Basically, the select inputs A0, A1 connect to the A0, A1 address bus signals of the CPU. The CS can be derived directly from the address bus using a linear select method or it can be connected to the output of a decoder, such as a Harris HD-6440 for larger systems.

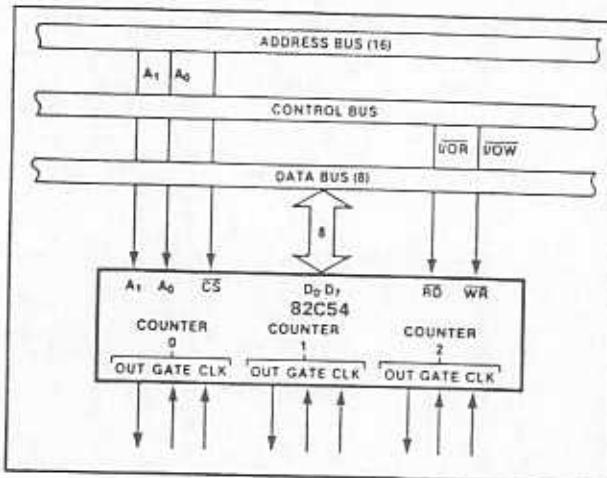


Figure 4. 82C54 System Interface

Operational Description

General

After power-up, the state of the 82C54 is undefined. The Mode, count value, and output of all Counters are undefined.

How each Counter operates is determined when it is programmed. Each Counter must be programmed before it can be used. Unused counters need not be programmed.

Programming The 82C54

Counters are programmed by writing a Control Word and then an initial count.

All Control Words are written into the Control Word Register, which is selected when A1, A0 = 11. The Control Word specifies which Counter is being programmed.

By contrast, initial counts are written into the Counters, not the Control Word Register. The A1, A0 inputs are used to select the Counter to be written. The format of the initial count is determined by the Control Word used.

82C54

Control Word Format

A1, A0 = 1, CS = 0, RD = 1, WR = 0

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
SC1	SC0	RW1	RW0	M2	M1	M0	BCD

SC — Select Counter:

SC1	SC0	Description
0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	Read-Back Command (See Read Operations)

RW — Read/Write:

RW1	RW0	Description
0	0	Counter Latch Command (see Read Operations)
0	1	Read/Write least significant byte only.
1	0	Read/Write most significant byte only.
1	1	Read/Write least significant byte first, then most significant byte.

M — MODE:

M2	M1	M0	Description
0	0	0	Mode 0
0	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

BCD:

Value	Description
0	Binary Counter 16-bits
1	Binary Coded Decimal (BCD) Counter (4 Decades)

NOTE: DON'T CARE BITS (X) SHOULD BE 0 TO INSURE COMPATIBILITY WITH FUTURE PRODUCTS.

Figure 5. Control Word Format

Write Operations

The programming procedure for the 82C54 is very flexible. Only two conventions need to be remembered:

1. For each Counter, the Control Word must be written before the initial count is written.
2. The initial count must follow the count format specified in the Control Word (least significant byte only, most significant byte only, or least significant byte and then most significant byte).

Since the Control Word Register and the three Counters have separate addresses (selected by the A1, A0 inputs), and each Control Word specifies the Counter it applies to (SC0, SC1 bits), no special instruction sequence is required. Any programming sequence that follows the conventions above is acceptable.

A new initial count may be written to a Counter at any time without affecting the Counter's programmed Mode in any way. Counting will be affected as described in the Mode definitions. The new count must follow the programmed count format.

If a Counter is programmed to read/write two-byte counts, the following precaution applies: A program must not transfer control between writing the first and second byte to another routine which also writes into that same Counter. Otherwise, the Counter will be loaded with an incorrect count.

Read Operations

It is often desirable to read the value of a Counter without disturbing the count in progress. This is easily done in the 82C54.

There are three possible methods for reading the Counters. The first is through the Read-Back command, which is explained later. The second is a simple read operation of the Counter, which is selected with the A1, A0 inputs. The only requirement is that the CLK input of the selected Counter must be inhibited by using either the GATE input or external logic. Otherwise, the count may be in process of changing when it is read, giving an undefined result.

	A ₁	A ₀		A ₁	A ₀
Control Word — Counter 0	1	1	Control Word — Counter 2	1	1
LSB of count — Counter 0	0	0	Control Word — Counter 1	1	1
MSB of count — Counter 0	0	0	Control Word — Counter 0	1	1
Control Word — Counter 1	1	1	LSB of count — Counter 2	1	0
LSB of count — Counter 1	0	1	MSB of count — Counter 2	1	0
MSB of count — Counter 1	0	1	LSB of count — Counter 1	0	1
Control Word — Counter 2	1	1	MSB of count — Counter 1	0	1
LSB of count — Counter 2	1	0	LSB of count — Counter 0	0	0
MSB of count — Counter 2	1	0	MSB of count — Counter 0	0	0
	A ₁	A ₀		A ₁	A ₀
Control Word — Counter 0	1	1	Control Word — Counter 1	1	1
Control Word — Counter 1	1	1	Control Word — Counter 0	1	1
Control Word — Counter 2	1	1	LSB of count — Counter 1	0	1
LSB of count — Counter 2	1	0	Control Word — Counter 2	1	1
LSB of count — Counter 1	0	1	LSB of count — Counter 0	0	0
LSB of count — Counter 0	0	0	MSB of count — Counter 1	0	1
MSB of count — Counter 0	0	0	LSB of count — Counter 2	1	0
MSB of count — Counter 1	0	1	MSB of count — Counter 0	0	0
MSB of count — Counter 2	1	0	MSB of count — Counter 2	1	0

NOTE: IN ALL FOUR EXAMPLES, ALL COUNTERS ARE PROGRAMMED TO READ/WRITE TWO-BYTE COUNTS.
THESE ARE ONLY FOUR OF MANY POSSIBLE PROGRAMMING SEQUENCES.

Figure 6. A Few Possible Programming Sequences

Counter Latch Command

The other method involves a special software command called the "Counter Latch Command". Like a Control Word, this command is written to the Control Word Register, which is selected when A₁, A₀ = 11. Also, like a Control Word, the SC0, SC1 bits select one of the three Counters, but two other bits, D₅ and D₄, distinguish this command from a Control Word.

The selected Counter's output latch (OL) latches the count when the Counter Latch Command is received. This count is held in the latch until it is read by the CPU (or until the Counter is reprogrammed). The count is then unlatched automatically and the OL returns to following the counting element (CE). This allows reading the contents of the Counters on the fly without affecting counting in progress. Multiple Counter Latch Commands may be used to latch more than one Counter. Each latched Counter's OL holds its count until read. Counter Latch Commands do not affect the programmed Mode of the Counter in any way.

A ₁ , A ₀ = 11; CS = 0; RD = 1; WR = 0																						
D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀															
SC1	SC0	0	0	X	X	X	X															
SC1, SC0 — specify counter to be latched																						
<table border="1"> <thead> <tr> <th>SC1</th> <th>SC0</th> <th>Counter</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>2</td> </tr> <tr> <td>1</td> <td>1</td> <td>Read-Back Command</td> </tr> </tbody> </table>								SC1	SC0	Counter	0	0	0	0	1	1	1	0	2	1	1	Read-Back Command
SC1	SC0	Counter																				
0	0	0																				
0	1	1																				
1	0	2																				
1	1	Read-Back Command																				
D ₅ , D ₄ — 00 designates Counter Latch Command																						
X — don't care																						
NOTE: DON'T CARE BITS (X) SHOULD BE 0 TO INSURE COMPATIBILITY WITH FUTURE PRODUCTS																						

Figure 7. Counter Latch Command Format

If a Counter is latched and then, some time later, latched again before the count is read, the second Counter Latch Command is ignored. The count read will be the count at the time the first Counter Latch Command was issued.

With either method, the count must be read according to the programmed format; specifically, if the Counter is programmed for two byte counts, two bytes must be read. The two bytes do not have to be read one right after the other; read or write or programming operations of other Counters may be inserted between them.

Another feature of the 82C54 is that reads and writes of the same Counter may be interleaved; for example, if the Counter is programmed for two byte counts, the following sequence is valid.

1. Read least significant byte.
2. Write new least significant byte.
3. Read most significant byte.
4. Write new most significant byte.

If a Counter is programmed to read or write two-byte counts, the following precaution applies: A program MUST NOT transfer control between reading the first and second byte to another routine which also reads from that same Counter. Otherwise, an incorrect count will be read.

Read-Back Command

The read-back command allows the user to check the count value, programmed Mode, and current state of the OUT pin and Null Count flag of the selected counter(s).

The command is written into the Control Word Register and has the format shown in Figure 8. The command applies to the counters selected by setting their corresponding bits D3, D2, D1 = 1.

A0, A1 = 11 C5 = 0 RD = 1 WR = 0							
D7	D6	D5	D4	D3	D2	D1	D0
t	1	COUNT	STATUS	CNT 2	CNT 1	CNT 0	0

D5: 0 - LATCH COUNT OF SELECTED COUNTER(S)
D4: 0 - LATCH STATUS OF SELECTED COUNTER(S)
D3: 1 - SELECT COUNTER 2
D2: 1 - SELECT COUNTER 1
D1: 1 - SELECT COUNTER 0
D0: RESERVED FOR FUTURE EXPANSION; MUST BE 0

Figure 8. Read-Back Command Format

The read-back command may be used to latch multiple counter output latches (OL) by setting the COUNT bit D5 = 0 and selecting the desired counter(s). This single command is functionally equivalent to several counter latch commands, one for each counter latched. Each counter's latched count is held until it is read (or the counter is reprogrammed). That counter is automatically unlatched when read, but other counters remain latched until they are read. If multiple count read-back commands are issued to the same counter without reading the count, all but the first are ignored; i.e., the count which will be read is the count at the time the first read-back command was issued.

The read-back command may also be used to latch status information of selected counter(s) by setting STATUS bit D4 = 0. Status must be latched to be read; status of a counter is accessed by a read from that counter.

The counter status format is shown in Figure 9. Bits D5 through D0 contain the counter's programmed Mode exactly as written in the last Mode Control Word. OUT-PUT bit D7 contains the current state of the OUT pin. This allows the user to monitor the counter's output via software, possibly eliminating some hardware from a system.

D7	D6	D5	D4	D3	D2	D1	D0
OUTPUT	NULL COUNT	RW1	RWD	M2	M1	M0	BCD

D7 = OUT PIN IS 1
0 = OUT PIN IS 0
D6 = NULL COUNT
0 = COUNT AVAILABLE FOR READING
D5-D0 = COUNTER PROGRAMMED MODE (SEE FIGURE 5)

Figure 9. Status Byte

NULL COUNT bit D6 indicates when the last count written to the counter register (CR) has been loaded into the counting element (CE). The exact time this happens depends on the Mode of the counter and is described in the Mode Definitions, but until the count is loaded into the counting element (CE), it can't be read from the counter. If the count is latched or read before this time, the count value will not reflect the new count just written. The operation of Null Count is shown in Figure 10.

THIS ACTION:	CAUSES:
A. WRITE TO THE CONTROL WORD REGISTER: ^[1]	NULL COUNT-1
B. WRITE TO THE COUNT REGISTER (CR): ^[2]	NULL COUNT-1
C. NEW COUNT IS LOADED INTO CE (CR-CE):	NULL COUNT=0
[1] ONLY THE COUNTER SPECIFIED BY THE CONTROL WORD WILL HAVE ITS NULL COUNT SET TO 1. NULL COUNT BITS OF OTHER COUNTERS ARE UNAFFECTED.	
[2] IF THE COUNTER IS PROGRAMMED FOR TWO-BYTE COUNTS (LEAST SIGNIFICANT BYTE THEN MOST SIGNIFICANT BYTE) NULL COUNT GOES TO 1 WHEN THE SECOND BYTE IS WRITTEN	

Figure 10. Null Count Operation

If multiple status latch operations of the counter(s) are performed without reading the status, all but the first are ignored; i.e., the status that will be read is the status of the counter at the time the first status read-back command was issued.

Both count and status of the selected counter(s) may be latched simultaneously by setting both COUNT and STATUS bits D5, D4 = 0. This is functionally the same as issuing two separate read-back commands at once, and the above discussions apply here also. Specifically, if multiple count and/or status read-back commands are issued to the same counter(s) without any intervening reads, all but the first are ignored. This is illustrated in Figure 11.

If both count and status of a counter are latched, the first read operation of that counter will return latched status, regardless of which was latched first. The next one or two reads (depending on whether the counter is programmed for one or two byte counts) return latched count. Subsequent reads return unlatched count.

CS	RD	WR	A ₁	A ₀	
0	1	0	0	0	Write into Counter 0
0	1	0	0	1	Write into Counter 1
0	1	0	1	0	Write into Counter 2
0	1	0	1	1	Write Control Word
0	0	1	0	0	Read from Counter 0
0	0	1	0	1	Read from Counter 1
0	0	1	1	0	Read from Counter 2
0	0	1	1	1	No-Operation (3-State)
1	X	X	X	X	No-Operation (3-State)
0	1	1	X	X	No-Operation (3-State)

Figure 12. Read/Write Operations Summary

Mode Definitions

The following are defined for use in describing the operation of the 82C54.

CLK PULSE: a rising edge, then a falling edge, in that order, of a Counter's CLK input.

TRIGGER: a rising edge of a Counter's Gate input.

COUNTER

LOADING: the transfer of a count from the CR to the CE (See "Functional Description")

Mode 0: Interrupt on Terminal Count

Mode 0 is typically used for event counting. After the

Control Word is written, OUT is initially low, and will remain low until the Counter reaches zero. OUT then goes high and remains high until a new count or a new Mode 0 Control Word is written to the Counter.

GATE = 1 enables counting; GATE = 0 disables counting. GATE has no effect on OUT.

After the Control Word and initial count are written to a Counter, the initial count will be loaded on the next CLK pulse. This CLK pulse does not decrement the count, so for an intial count of N, OUT does not go high until N + 1 CLK pulses after the initial count is written.

If a new count is written to the Counter it will be loaded on the next CLK pulse and counting will continue from the new count. If a two-byte count is written, the following happens:

- 1 - Writing the first byte disables counting. Out is set low immediately (no clock pulse required).
- 2 - Writing the second byte allows the new count to be loaded on the next CLK pulse.

This allows the counting sequence to be synchronized by software. Again OUT does not go high until N + 1 CLK pulses after the new count of N is written.

If an initial count is written while GATE = 0, it will still be loaded on the next CLK pulse. When GATE goes high, OUT will go high N CLK pulses later; no CLK pulse is needed to load the Counter as this has already been done.

Command								Description	Result
D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀		
1	1	0	0	0	0	1	0	Read back count and status of Counter 0	Count and status latched for Counter 0
1	1	1	0	0	1	0	0	Read back status of Counter 1	Status latched for Counter 1
1	1	1	0	1	1	0	0	Read back status of Counters 2, 1	Status latched for Counter 2, but not Counter 1
1	1	0	1	1	0	0	0	Read back count of Counter 2	Count latched for Counter 2
1	1	0	0	0	1	0	0	Read back count and status of Counter 1	Count latched for Counter 1, but not status
1	1	1	0	0	0	1	0	Read back status of Counter 1	Command ignored, status already latched for Counter 1

Figure 11. Read-Back Command Example

Mode 1: Hardware Retriggerable One-Shot

OUT will be initially high. OUT will go low on the CLK pulse following a trigger to begin the one-shot pulse, and will remain low until the Counter reaches zero. OUT will then go high and remain high until the CLK pulse after the next trigger.

After writing the Control Word and initial count, the Counter is armed. A trigger results in loading the Counter and setting OUT low on the next CLK pulse, thus starting the one-shot pulse N CLK cycles in duration. The one-shot is retriggerable, hence OUT will remain low for N CLK

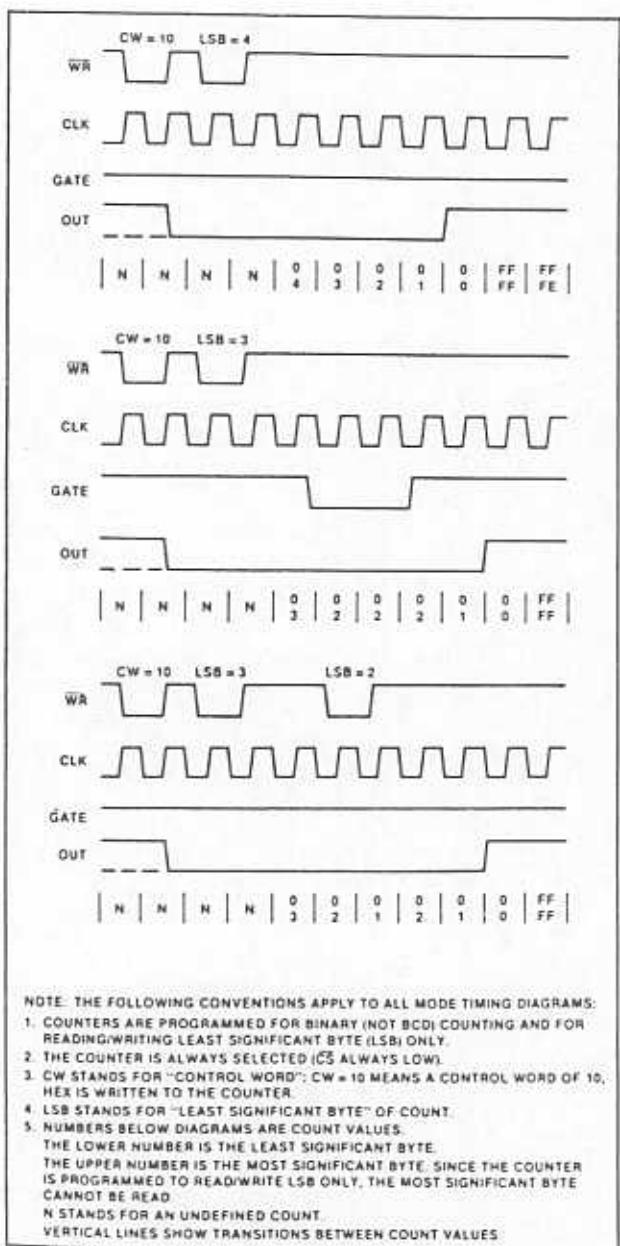


Figure 13. Mode 0

pulses after any trigger. The one-shot pulse can be repeated without rewriting the same count into the counter. GATE has no effect on OUT.

If a new count is written to the Counter during a one-shot pulse, the current one-shot is not affected unless the Counter is retriggered. In that case, the Counter is loaded with the new count and the one-shot pulse continues until the new count expires.

Mode 2: Rate Generator

This Mode functions like a divide-by-N counter. It is typically used to generate a Real Time Clock interrupt. OUT will initially be high. When the initial count has decremented to 1, OUT goes low for one CLK pulse. OUT then goes high again, the Counter reloads the initial count and the process is repeated. Mode 2 is periodic; the same sequence is repeated indefinitely. For an initial count of N, the sequence repeats every N CLK cycles.

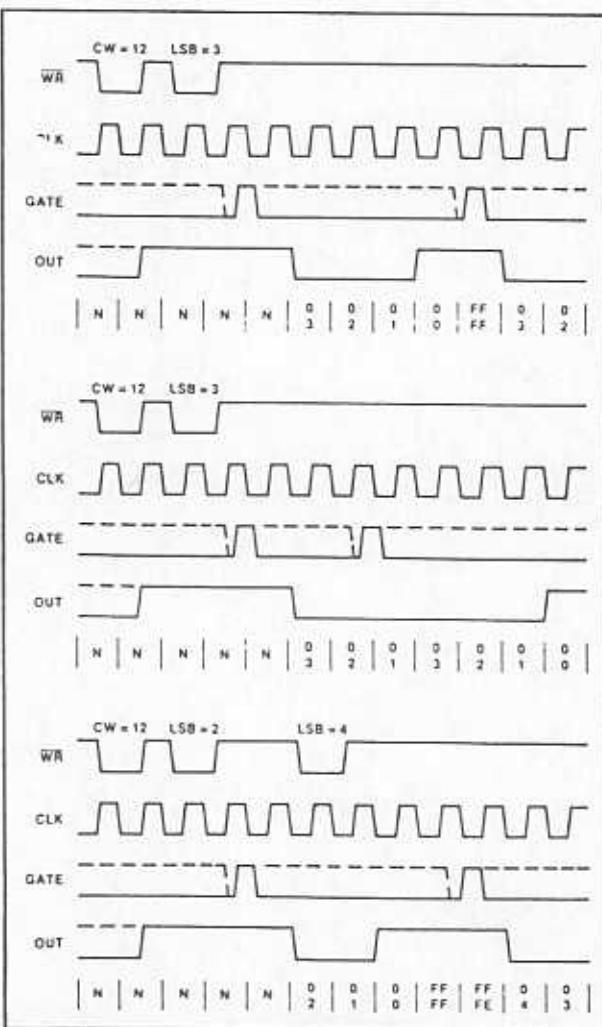


Figure 14. Mode 1

GATE = 1 enables counting; GATE = 0 disables counting. If GATE goes low during an output pulse, OUT is set high immediately. A trigger reloads the Counter with the initial count on the next CLK pulse; OUT goes low N CLK pulses after the trigger. Thus the GATE input can be used to synchronize the Counter.

After writing a Control Word and initial count, the Counter will be loaded on the next CLK pulse. OUT goes low N CLK pulses after the initial count is written. This allows the Counter to be synchronized by software also.

Writing a new count while counting does not affect the current counting sequence. If a trigger is received after writing a new count but before the end of the current period, the Counter will be loaded with the new count on the next CLK pulse and counting will continue from the new count. Otherwise, the new count will be loaded at the end of the current counting cycle.

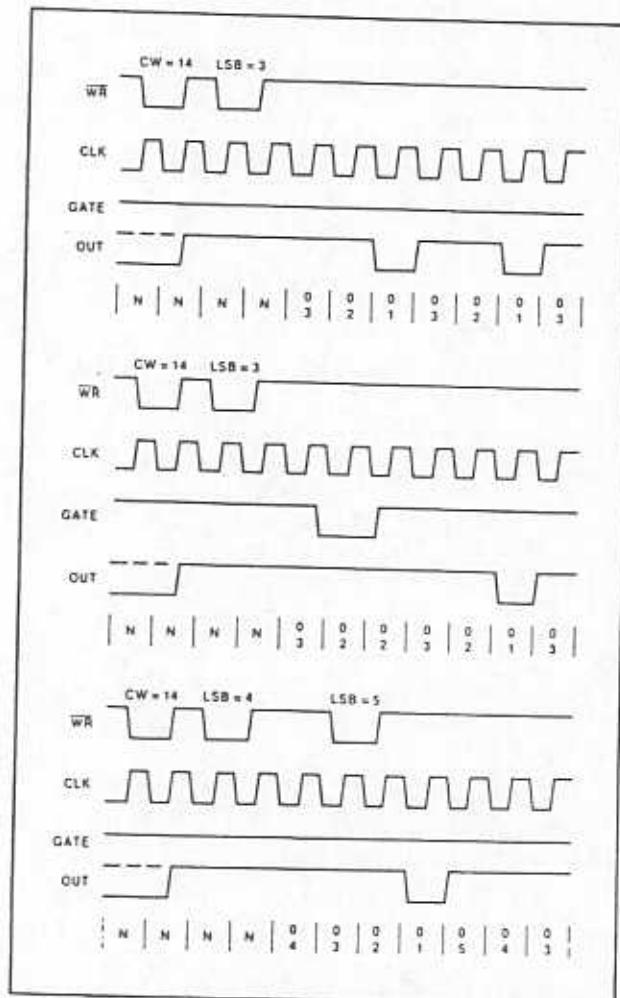


Figure 15. Mode 2

Mode 3: Square Wave Mode

Mode 3 is typically used for Baud rate generation. Mode 3 is similar to Mode 2 except for the duty cycle of OUT. OUT will initially be high. When half the initial count has expired, OUT goes low for the remainder of the count. Mode 3 is periodic; the sequence above is repeated indefinitely. An initial count of N results in a square wave with a period of N CLK cycles.

GATE = 1 enables counting; GATE = 0 disables counting. If GATE goes low while OUT is low, OUT is set high immediately; no CLK pulse is required. A trigger reloads the Counter with the initial count on the next CLK pulse. Thus the GATE input can be used to synchronize the Counter.

After writing a Control Word and initial count, the Counter will be loaded on the next CLK pulse. This allows the Counter to be synchronized by software also.

Writing a new count while counting does not affect the

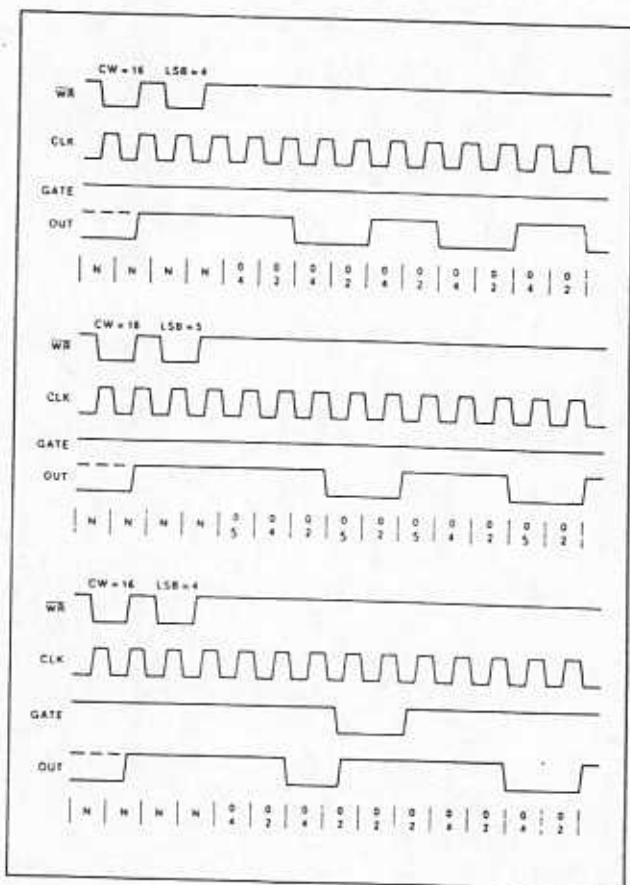


Figure 16. Mode 3

current counting sequence. If a trigger is received after writing a new count but before the end of the current half-cycle of the square wave, the Counter will be loaded with the new count on the next CLK pulse and counting will continue from the new count. Otherwise, the new count will be loaded at the end of the current half-cycle.

Mode 3 is implemented as follows:

EVEN COUNTS: OUT is initially high. The initial count is loaded on one CLK pulse and then is decremented by two on succeeding CLK pulses. When the count expires OUT changes value and the Counter is reloaded with the initial count. The above process is repeated indefinitely.

ODD COUNTS: OUT is initially high. The initial count is loaded on one CLK pulse, decremented by one on the next CLK pulse, and then decremented by two on succeeding CLK pulses. When the count expires, OUT goes low and the counter is reloaded with the initial count. The count is decremented by three on the next CLK pulse, and then by two on succeeding CLK pulses. When the count expires, OUT goes high again and the counter is reloaded with the initial count. The above process is repeated indefinitely. So for odd counts, OUT will be high for $(N+1)/2$ counts and low for $(N-1)/2$ counts.

Mode 4: Software Triggered Mode

OUT will be initially high. When the initial count expires, OUT will go low for one CLK pulse and then go high again. The counting sequence is "Triggered" by writing the initial count.

GATE = 1 enables counting; GATE = 0 disables counting. GATE has no effect on OUT.

After writing a Control Word and initial count, the Counter will be loaded on the next CLK pulse. This CLK pulse does not decrement the count, so for an initial count of N, OUT does not strobe low until $N + 1$ CLK pulses after the initial count is written.

If a new count is written during counting, it will be loaded on the next CLK pulse and counting will continue from the new count. If a two-byte count is written, the following happens:

1. Writing the first byte has no effect on counting.
2. Writing the second byte allows the new count to be loaded on the next CLK pulse.

This allows the sequence to be "retriggered" by software. OUT strobos low $N + 1$ CLK pulses after the new count of N is written.

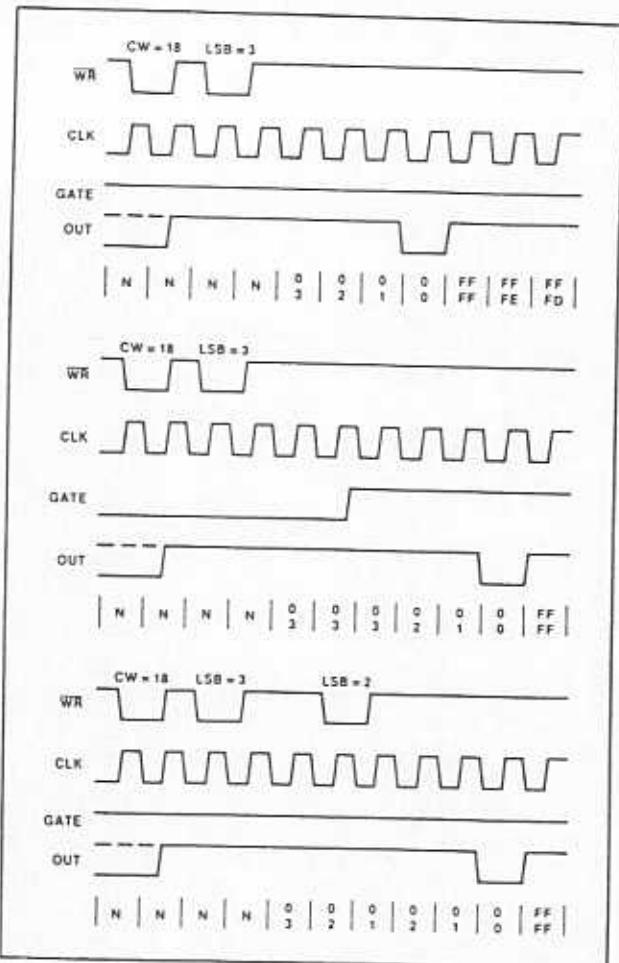


Figure 17. Mode 4

Mode 5: Hardware Triggered Strobe (Retriggerable)

OUT will initially be high. Counting is triggered by a rising edge of GATE. When the initial count has expired, OUT will go low for one CLK pulse and then go high again.

After writing the Control Word and initial count, the counter will not be loaded until the CLK pulse after a trigger. This CLK pulse does not decrement the count, so for an initial count of N, OUT does not strobe low until $N+1$ CLK pulses after trigger.

A trigger results in the Counter being loaded with the initial count on the next CLK pulse. The counting sequence is retriggerable. OUT will not strobe low for $N + 1$ CLK pulses after any trigger. GATE has no effect on OUT.

If a new count is written during counting, the current counting sequence will not be affected. If a trigger occurs after the new count is written but before the current count

expires, the Counter will be loaded with the new count on the next CLK pulse and counting will continue from there.

Operation Common to all Modes

Programming

When a Control Word is written to a Counter, all Control Logic is immediately reset and OUT goes to a known initial state; no CLK pulses are required for this.

Gate

The GATE input is always sampled on the rising edge of CLK. In Modes 0, 2, 3, and 4 the Gate input is level sensitive, and the logic level is sampled on the rising edge of CLK. In Modes 1, 2, 3, and 5 the GATE input is rising-edge sensitive. In these Modes, a rising edge of Gate (trigger) sets an edge-sensitive flip-flop in the Counter. This flip-flop is then sampled on the next rising edge of CLK.

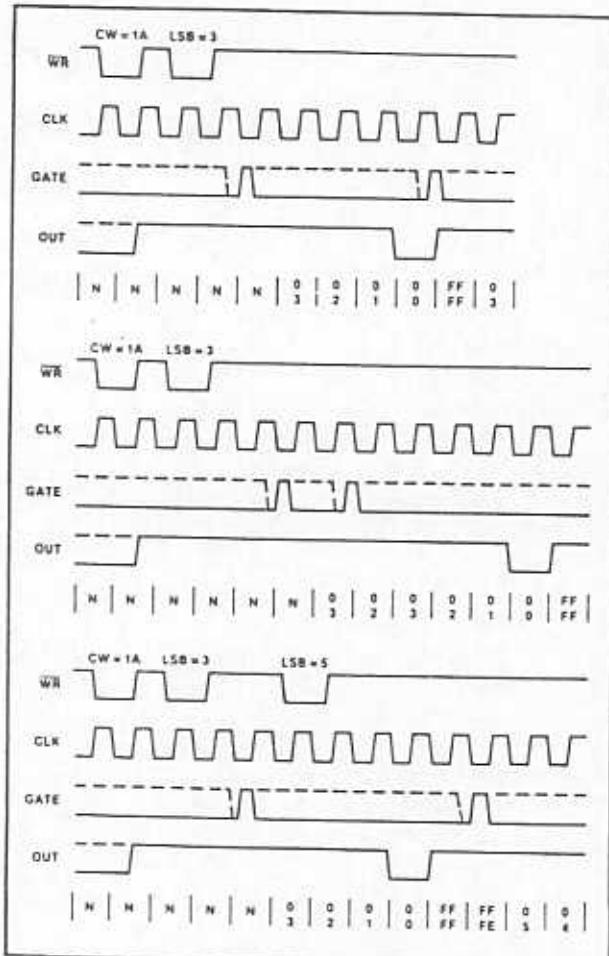


Figure 18. Mode 5

The flip-flop is reset immediately after it is sampled. In this way, a trigger will be detected no matter when it occurs - a high logic level does not have to be maintained until the next rising edge of CLK. Note that in Modes 2 and 3, the GATE input is both edge- and level-sensitive.

Counter

New counts are loaded and Counters are decremented on the falling edge of CLK.

The largest possible initial count is 0; this is equivalent to 2^{16} for binary counting and 10^4 for BCD counting.

The Counter does not stop when it reaches zero. In Modes 0, 1, 4, and 5 the Counter "wraps around" to the highest count, either FFFF hex for binary counting or 9999 for BCD counting, and continues counting. Modes 2 and 3 are periodic; the Counter reloads itself with the initial count and continues counting from there.

Signal Status Modes	Low Or Going Low	Rising	High
0	Disables counting	---	Enables counting
1	---	1) Initiates counting 2) Resets output after next clock	---
2	1) Disables counting 2) Sets output immediately high	Initiates counting	Enables counting
3	1) Disables counting 2) Sets output immediately high	Initiates counting	Enables counting
4	Disables counting	---	Enables counting
5	---	Initiates counting	---

Figure 19. Gate Pin Operations Summary

Mode	Min Count	Max Count
0	1	0
1	1	0
2	2	0
3	2	0
4	1	0
5	1	0

NOTE: 0 IS EQUIVALENT TO 2^{16} FOR BINARY COUNTING AND 10^4 FOR BCD COUNTING

Figure 20. Minimum and Maximum Initial Counts

Specifications

ABSOLUTE MAXIMUM RATINGS

Supply Voltage	+8.0Volts
Input, Output or I/O Voltage Applied	GND -0.5V to VCC +0.5V
Storage Temperature Range	-65°C to +150°C
Maximum Package Power Dissipation	1 Watt

CAUTION: Stresses above those listed in the "ABSOLUTE MAXIMUM RATINGS" may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

RECOMMENDED OPERATING CONDITIONS

Operating Voltage Range	+4.5V to +5.5V
Operating Temperature Ranges:	
Commercial	0°C to +70°C
Industrial	-40°C to +85°C
Military	-55°C to +125°C

D.C. ELECTRICAL CHARACTERISTICS

VCC = 5.0V±10%; TA = 0°C to +70°C (C82C54); TA = -40°C to +85°C (I82C54); TA = -55°C to +125°C (M82C54)

SYMBOL	PARAMETER	MIN	MAX	UNITS	TEST CONDITIONS
VIH	Logical One Input Voltage	2.0		V	C82C54, I82C54 M82C54
	Input Voltage	2.2		V	
VIL	Logical Zero Input Voltage		0.8	V	
	Output High Voltage	3.0		V	
VOH		VCC - 0.4		V	IOH = -2.5mA IOH = -100 μA
	Output Low Voltage		0.4	V	
IIL	Input Leakage Current	-1.0	+1.0	μA	0V ≤ VIN ≤ VCC
IO	Output Leakage Current	-10.0	+10.0	μA	0V ≤ VO ≤ VCC
ICCSB	Standby Power Supply Current		10	μA	VCC = 5.5V VIN = VCC or GND OUTPUTS OPEN
ICCOP	Operating Power Supply Current		10	mA	VCC = 5.5V CLK FREQ = 8MHz OUTPUTS OPEN

CAPACITANCE

TA = 25°C; VCC = GND = 0V; VIN = +5V or GND

SYMBOL	PARAMETER	TYPICAL	UNITS	TEST CONDITIONS
CIN	Input Capacitance	5	pF	FREQ = 1MHz Unmeasured pins returned to GND
COUT	Output Capacitance	15	pF	
CI/O	I/O Capacitance	20	pF	

A.C. CHARACTERISTICS

$V_{CC} = +5V \pm 10\%$; $TA = 0^{\circ}\text{C}$ to $+70^{\circ}\text{C}$ (C82C54); $TA = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ (I82C54); $TA = -55^{\circ}\text{C}$ to $+125^{\circ}\text{C}$ (M82C54)

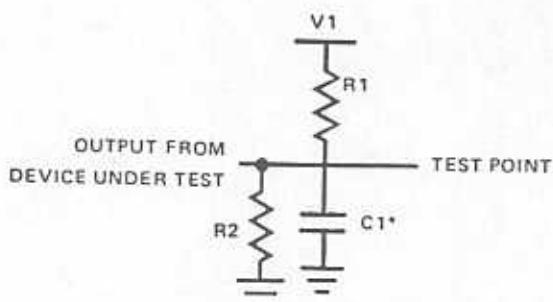
Bus Parameters

READ CYCLE		82C54		TEST CONDITIONS
SYMBOL	PARAMETER	MIN	MAX	
TAR	Address Stable Before RD	30		ns
TSR	CS Stable Before RD	0		ns
TRA	Address Hold Time After RD	0		ns
TRR	RD Pulse Width	150		ns
TRD	Data Delay from RD		120	ns
TAD	Data Delay from Address		210	ns
TDF	RD to Data Floating	5		ns
TRV	Command Recovery Time	200	85	ns

WRITE CYCLE				TEST CONDITIONS
SYMBOL	PARAMETER	MIN	MAX	
TAW	Address Stable Before WR	0		ns
TSW	CS Stable Before WR	0		ns
TWA	Address Hold Time WR	0		ns
TWW	WR Pulse Width	95		ns
TDW	Data Setup Time Before WR	140		ns
TWD	Data Hold Time After WR	25		ns
TRV	Command Recovery Time	200		ns

CLOCK AND GATE				TEST CONDITIONS
SYMBOL	PARAMETER	MIN	MAX	
TCLK	Clock Period	125	DC	ns
TPWH	High Pulse Width	60		1
TPWL	Low Pulse Width	60		ns
TR	Clock Rise Time	60	25	ns
TF	Clock Fall Time		25	ns
TGW	Gate Width High	50		ns
TGL	Gate Width Low	50		ns
TGS	Gate Setup Time to CLK	50		ns
TGH	Gate Hold Time After CLK	50		ns
TOD	Output Delay from CLK		150	ns
TODG	Output Delay from Gate		120	ns
TWO	OUT Delay from Mode Write		260	ns

A.C. Test Circuits

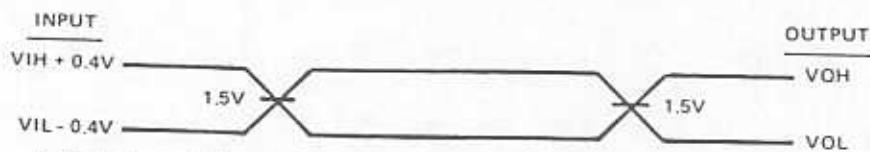


*Includes stray and jig capacitance

TEST CONDITION	V1	R1	R2	C1
1	1.7V	523	OPEN	150pf
2	5.0V	2K	1.7K	50pf

TEST CONDITION DEFINITION TABLE

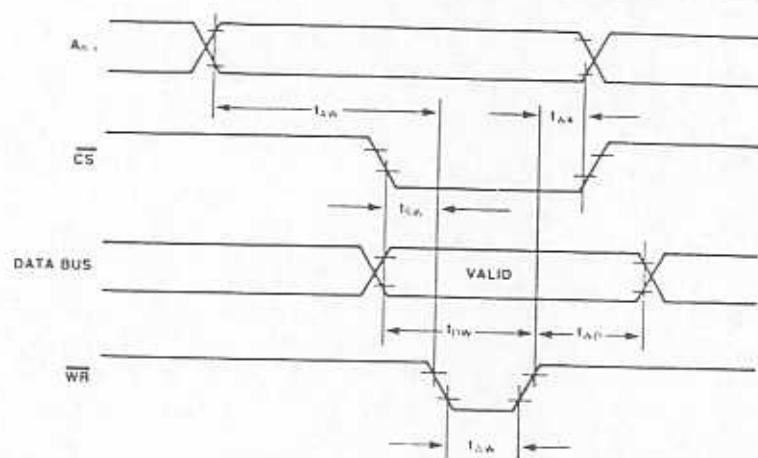
A.C. Testing Input, Output Waveform



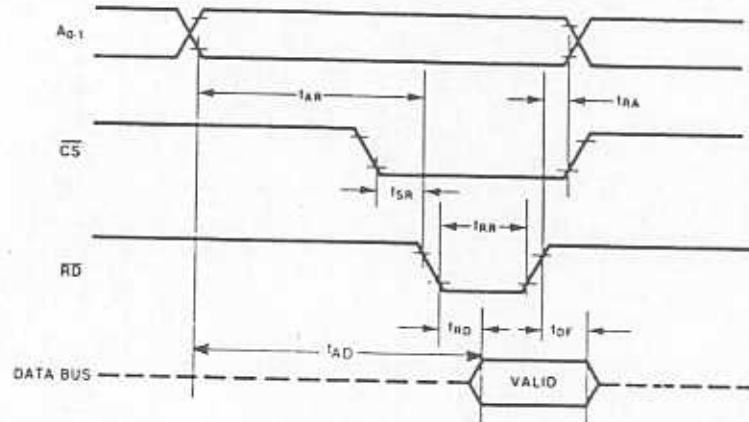
A.C. Testing: All input signals must switch between $VIL - 0.4V$ and $VIH + 0.4V$. Input rise and fall times are driven at 1ns/V.

Waveforms

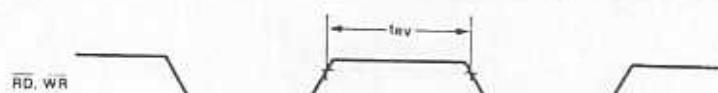
WRITE



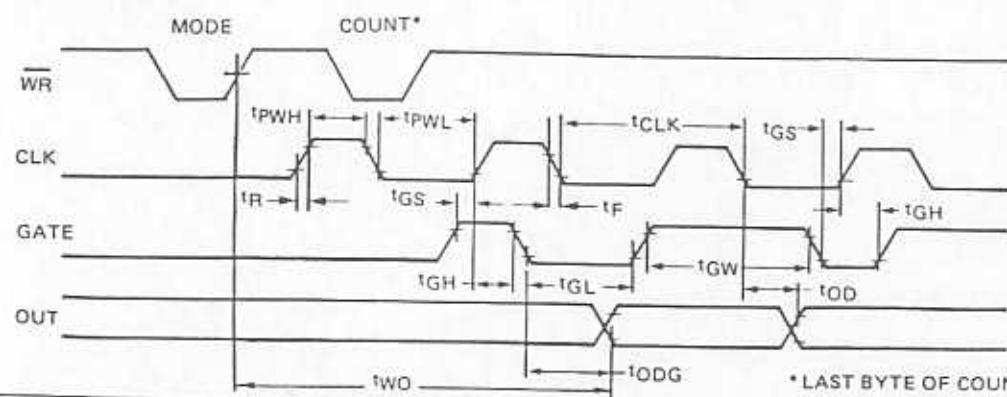
READ



RECOVERY

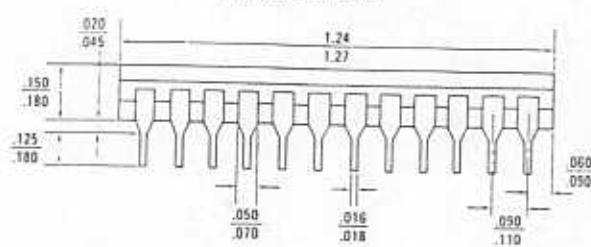


CLOCK AND GATE

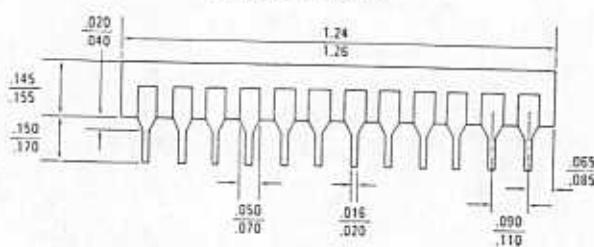


Packaging

24 LEAD CERDIP

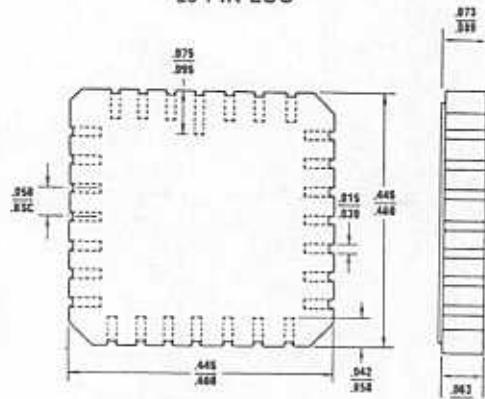


24 LEAD PLASTIC



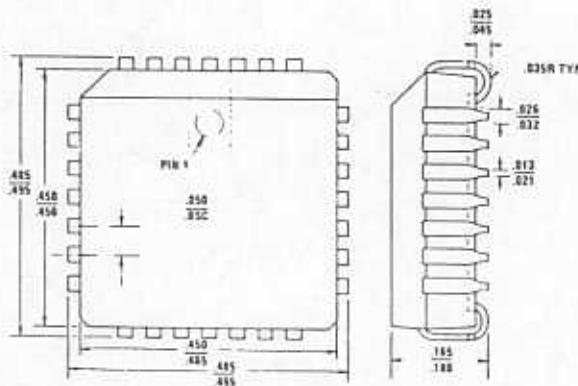
NOTE: 1) Dimensions are: MIN.
MAX.
2) All Dimensions in inches

28 PIN LCC



TOP VIEW

28 PIN PLCC

*Ordering Information*

M
TEMPERATURE
RANGE

C - Commercial
I - Industrial
M - Military
X - 25°C

D
PACKAGE
TYPE

P - Plastic
D - Ceramic
X - Chip Form
R - Leadless
Chip Carrier
S - Plastic Leaded
Chip Ceramic

82C54

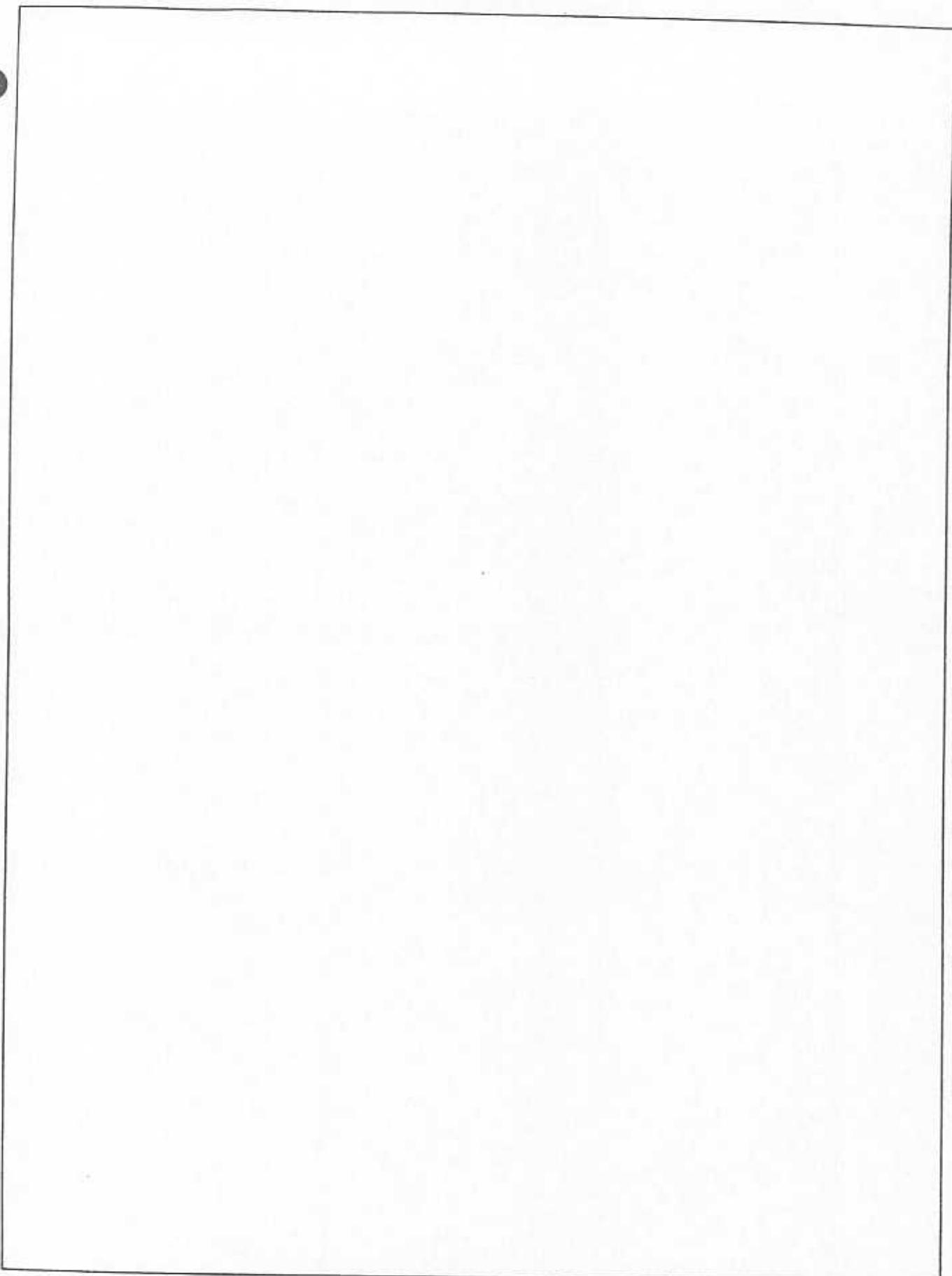
PART
NUMBER

/B

B: Dash B Hi-Rel Program
+: Plus Program - Industrial
Temperature w/160 hr. burn-in

NOTICE: Harris Semiconductor's products are sold by description only. Harris Semiconductor reserves the right to make changes in circuit design and/or specifications at any time without notice. Accordingly, the reader is cautioned to verify that data sheets are current before placing orders.

Notes



A large, empty rectangular box with a thin black border, intended for handwritten notes. The box occupies most of the page below the title.

Notes

FOR YOUR INFORMATION, OUR TRACTIC IS EASY.

OEM Sales

Alabama P.O. Box 7066 Huntsville, AL 35807 (205) 837-8886	Colorado Suite 206 51 Inverness Dr., East Englewood, CO 80112 (303) 799-8600	Massachusetts Suite 3A 83 Cambridge St. Burlington, MA 01803 (617) 273-5942	Pennsylvania Suite 1101 995 Old Eagle School Rd. Wayne, PA 19087 (215) 687-6680
Arizona Suite 250 1717 E. Morten Ave. Phoenix, AZ 85020 (602) 870-0080	Florida * P.O. Box 1239 Melbourne, FL 32901 (305) 724-3576	Minnesota Suite 703 2850 Metro Drive Minneapolis, MN 55420 (612) 854-3558	Texas * Suite 110 17120 Dallas Parkway Dallas, TX 75248 (214) 248-3237
California Suite 320 1503 South Coast Drive Costa Mesa, CA 92626 (714) 540-2176	Suite 215 3890 W. Commercial Blvd. Ft. Lauderdale, FL 33309 (305) 739-0016	North Carolina P.O. Box 31747 Raleigh, NC 27622 (919) 847-8985	Washington 33919 9th Ave., South Federal Way, WA 98003 (206) 838-4878
* Suite C100 883 Sierlin Road Mountain View, CA 94043 (415) 964-6443	875 Johnson Ferry Road N.E. Atlanta, GA 30342 (404) 256-4000	New York * Suite 426 555 Broadhollow Road Melville, NY 11747 (516) 249-4500	Suite 250 500 W. 8th Street Vancouver, WA 98660 (503) 283-7027
* Suite 205 6400 Canoga Avenue Woodland Hills, CA 91367 (818) 992-0686	Illinois * Suite 300 6400 Shafer Court Rosemont, IL 60018 (312) 692-4960	Ohio Suite 180 425 Metro Place, North Dublin, OH 43017 (614) 766-4111	

International Sales:

Sales Offices

ENGLAND * Harris/MHS Semiconductor Sales Ltd. Eskdale Road Winnersh Triangle, Workingham RG11 5TR Berkshire, United Kingdom Tel: 44-734-698787 TWX: 648-174	SWEDEN Harris/MHS Semiconductor Sales AB Stockholmsvagen 116 S-183 38 Taaby Tel: 46-08-7921240 TWX: 14867	Eltek Tel: (33) 1-34771616 EPROM Tel: (33) 91029776 Feutrier Tel: (33) 77746733 RTF Tel: (33) 1-46641101 Spetelec Tel: (33) 1-46865665	REPUBLIC OF S. AFRICA Advanced Semi Devices Tel: (0027)-11-8025820 Allied El. Ltd. Tel: (0027) 11-528661
FRANCE Matra-Harris Semiconducteurs Les Quadrants 3 Ave du Centre 78182 Saint-Quentin-en-Yvelines Tel: 33-1-30428272 TWX: 697317	WEST GERMANY * Harris/MHS Semiconductor Sales GmbH Erluterstrasse 29 D-8057 Eching Tel: 37089-3191035 TWX: 524126	WEST GERMANY Alfred Neye Enatechnik Tel: (49) 04106-6121 Distributors AUSTRIA Transistor Vertriebsges. m.b.H. & Co. KG Tel: (43) 0222-829401	SPAIN Selco S. A. Tel: (0034) 1-4094634
MHS Electronic Center MHS BP 942 44075 Nantes Cedex/France Tel: (33) 40303030 TWX: 711930 MHS Locazirst 5 Chemin des Pres 38240 Meylan Tel: (33) 76903090	BELGIUM MCA-Tronix BRAZIL Hitech Com. & Ind. Ltda. Tel: 11-533-9566 DENMARK Ditz Schweitzer A.S. Tel: (45) 02-453044	WEST GERMANY Kontron Halbleiter Tel: (49) 089-319010 Spoerle Electronics Tel: (49) 06103-3040	SWEDEN A B Betoma Tel: (46) 08-820280 Fertronic A B Tel: (46) 08-830060
ITALY Harris/MHS Semiconductor Sales SRL I-20092 Cinisello Balsamo Via Fratelli Cracchi 48 Milano Tel: 39-02-6188282 TWX: 324019	FINLAND Yleiselektronika OY Tel: (358) 090-5621122 FRANCE A2M Tel: (33) 1-3959113 Aimex Tel: (33) 1-46662112	ISRAEL MRBD Ltd. Tel: (00972) 3-736479/732624 ITALY Kontron SpA Tel: 02-50721 NETHERLANDS Techmation Electronics B. V.	SWITZERLAND Stoltz A. G. Tel: (0041) 56840151 UNITED KINGDOM AND IRELAND Elite Semiconductors Ltd. Tel: (44) 080-131155 Macro Marketing Ltd. Tel: (44) 062886-4422 Radio Resistor Electronics Ltd. Tel: (44) 0234-47188 Semiconductor Die Processing Ltd. Tel: (44) 0429-33721 Stewart Electronics Ltd. Tel: (44) 0555-892393 Thame Components Ltd. Tel: (44) 084421-4561 Trim Electronics Ltd. Tel: (353) 046-36263

International Sales: Far East

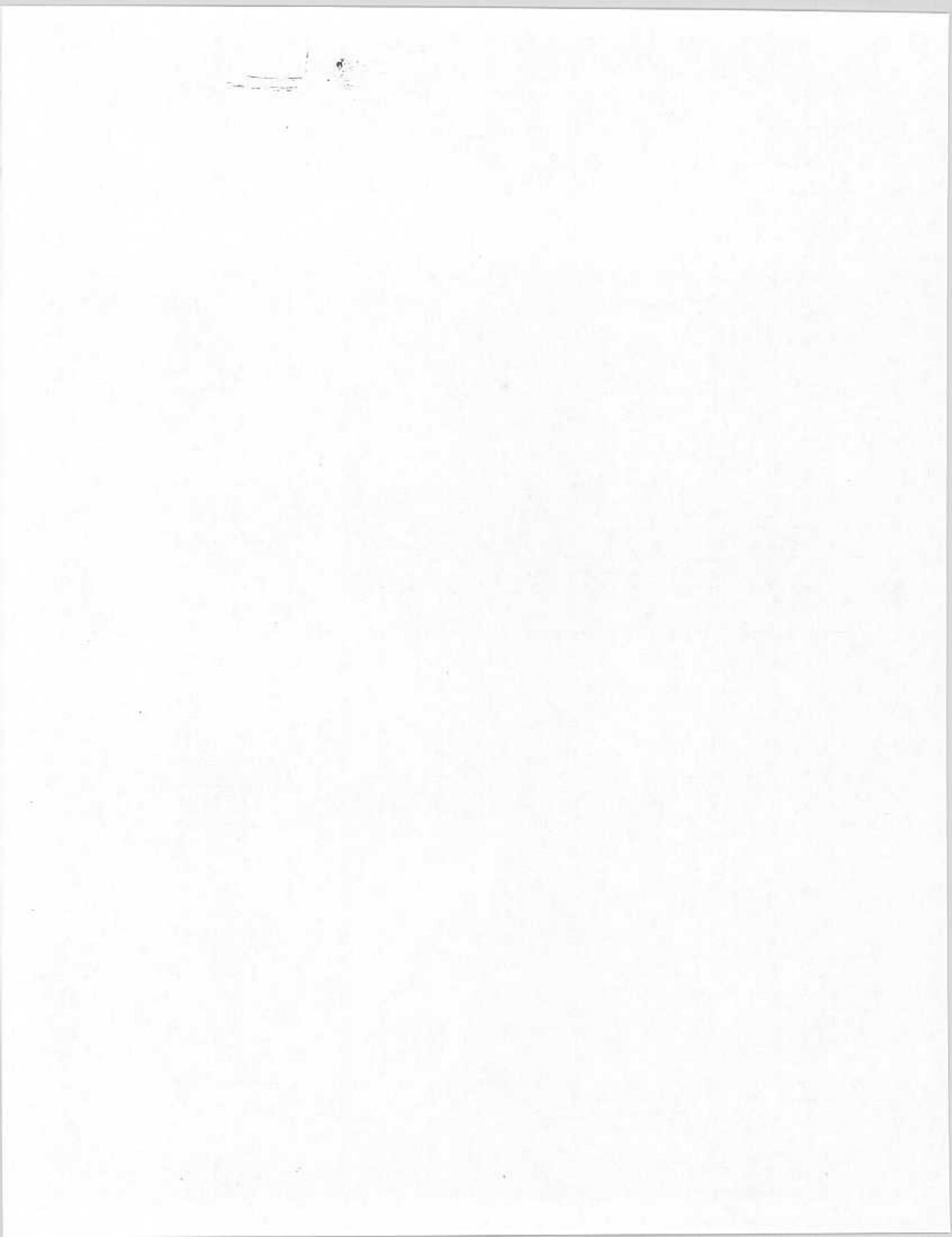
Sales Offices

Harris K. K. Shinjuku NS Bldg. Box 6153 2-4-1, Nishi-Shinjuku Shinjuku-ku Tokyo, 160 Japan Tel.: 03-485-1031 TWX: 326562	Far East Distributors AUSTRALIA VSI Electronics NEW ZEALAND Professional Electronics Ltd. HONG KONG Karin Elec. Supplies Co. Ltd.	JAPAN Japan Macnics Corp. Hakuto Co., Ltd. Micron Inc. Ryoyo Electric Corporation SINGAPORE Desner Electronic (Far East)	TAIWAN Leadtron Industrial Inc. INDIA Fegu Electronics KOREA TESS-KO Enterprise
* Field Applications Assistance Available			



SEMICONDUCTOR DIGITAL DIVISION

80C88 MICROPROCESSOR



Features

- COMPATIBLE WITH NMOS 8088
- DIRECT SOFTWARE COMPATIBILITY WITH 80C86, 8086, 8088
- 8 BIT DATA BUS INTERFACE
- 16 BIT INTERNAL ARCHITECTURE
- COMPLETELY STATIC DESIGN
 - DC - 5 MHz (80C88)
 - DC - 4 MHz (80C88-4)
- LOW POWER OPERATION
 - ICCSB = 500 μ A MAXIMUM
 - ICCOP = 10mA/MHz
- 1 MBYTE OF DIRECT MEMORY ADDRESSING CAPABILITY
- 24 OPERAND ADDRESSING MODES
- BIT, BYTE, WORD, AND BLOCK MOVE OPERATIONS
- 8 AND 16 BIT SIGNED/UNSIGNED ARITHMETIC
- BUS-HOLD CIRCUITRY ELIMINATES PULL-UP RESISTORS
- SCALED SAJI IV CMOS PROCESS
- SINGLE 5V POWER SUPPLY
- COMMERCIAL, INDUSTRIAL AND MILITARY TEMPERATURE RANGES

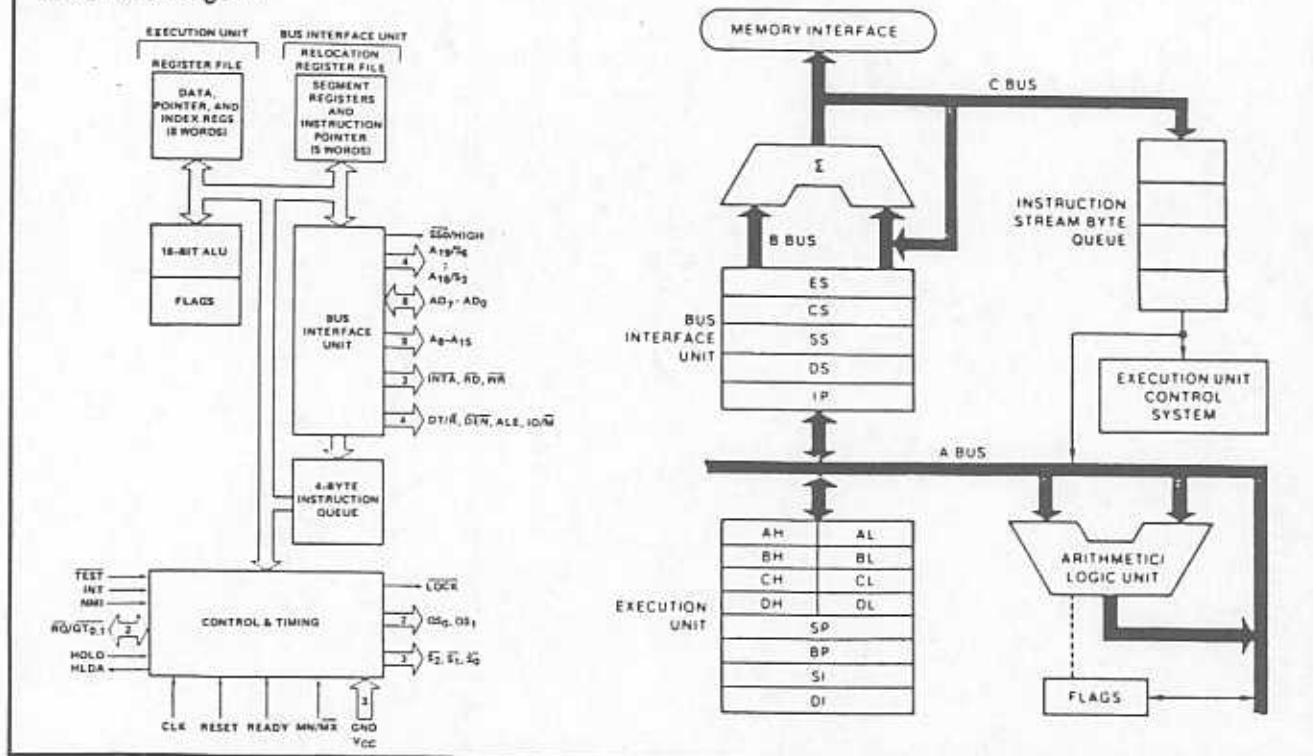
Pinout

		MIN MODE	MAX MODE
GND	1	40	VCC
A14	2	39	A15
A13	3	38	A16/S3
A12	4	37	A17/S4
A11	5	36	A18/S5
A10	6	35	A19/S6
A9	7	34	SS0 (HIGH)
A8	8	33	MNIMX
AD7	9	32	RD
AD6	10	BDC88 CPU	RD/GTO
AD5	11	31	HOLD (RD/GTO)
AD4	12	30	HLDA (RD/GTO)
AD3	13	29	WR (LOCK)
AD2	14	28	IO/M (S2)
AD1	15	27	DT/R (S1)
AD0	16	26	DEN (S0)
NMI	17	25	ALE (QS0)
INTR	18	24	INTA (QS1)
CLK	19	23	TEST
GND	20	22	READY
		21	RESET

Description

The Harris 80C88 high performance 8/16 bit CMOS CPU is manufactured using a self-aligned silicon gate CMOS process (Scaled SAJI IV). Two modes of operation, MINimum for small systems and MAXimum for larger applications such as multiprocessing, allow user configuration to achieve the highest performance level. Full TTL compatibility and industry-standard operation allow use of existing NMOS 8088 hardware and Harris CMOS 80C86 peripherals. Complete software compatibility with the 80C86, 8086 and 8088 microprocessors allows use of existing software in new designs.

Functional Diagram



CAUTION: Electronic devices are sensitive to electrostatic discharge. Proper I.C. handling procedures should be followed.

Copyright © Harris Corporation 1984

Pin Description

The following pin function descriptions are for 80C88 systems in either minimum or maximum mode. The "local bus" in these

descriptions is the direct multiplexed bus interface connection to the 80C88 (without regard to additional bus buffers).

SYMBOL	PIN NUMBER	TYPE	NAME AND FUNCTION
AD7-AD0	9-16	I/O	Address Data Bus: These lines constitute the time multiplexed memory/I/O address (T1) and data (T2, T3, Tw, and T4) bus. These lines are active HIGH and are held at high impedance to the last valid logic level during interrupt acknowledge and local bus "hold acknowledge" or "grant sequence".
A15-A8	2-8, 39	O	Address Bus: These lines provide address bits 8 through 15 for the entire bus cycle (T1-T4). These lines do not have to be latched by ALE to remain valid. A15-A8 are active HIGH and are held at high impedance to the last valid logic level during interrupt acknowledge and local bus "hold acknowledge" or "grant sequence".
A19/S6, A18/S5, A17/S4, A16/S3	35 36 37 38	O	<p>Address/Status: During T1, these are the four most significant address lines for memory operations. During I/O operations, these lines are LOW. During memory and I/O operations, status information is available on these lines during T2, T3, Tw, and T4. S6 is always low. The status of the interrupt enable flag bit (S5) is updated at the beginning of each clock cycle. S4 and S3 are encoded as shown.</p> <p>This information indicates which segment register is presently being used for data accessing.</p> <p>These lines are held at high impedance to the last valid logic level during local bus "hold acknowledge" or "grant sequence".</p>
\overline{RD}	32	O	<p>Read: Read strobe indicates that the processor is performing a memory or I/O read cycle, depending on the state of the IO/M pin or S2. This signal is used to read devices which reside on the 80C88 local bus. \overline{RD} is active LOW during T2, T3 and Tw of any read cycle, and is guaranteed to remain HIGH in T2 until the 80C88 local bus has floated.</p> <p>This line is held internally to a high impedance logic one state during "hold acknowledge" or "grant sequence".</p>
READY	22	I	READY: is the acknowledgment from the addressed memory or I/O device that it will complete the data transfer. The RDY signal from memory or I/O is synchronized by the 82C84A clock generator to form READY. This signal is active HIGH. The 80C88 READY input is not synchronized. Correct operation is not guaranteed if the set up and hold times are not met.
INTR	18	I	Interrupt Request: is a level triggered input which is sampled during the last clock cycle of each instruction to determine if the processor should enter into an interrupt acknowledge operation. A subroutine is vectored to via an interrupt vector lookup table located in system memory. It can be internally masked by software resetting the interrupt enable bit. INTR is internally synchronized. This signal is active HIGH.
TEST	23	I	TEST: input is examined by the "wait for test" instruction. If the TEST input is LOW, execution continues, otherwise the processor waits in an "idle" state. This input is synchronized internally during each clock cycle on the leading edge of CLK.
NMI	17	I	NON-MASKABLE INTERRUPT: is an edge triggered input which causes a type 2 interrupt. A subroutine is vectored to via an interrupt vector lookup table located in system memory. NMI is not maskable internally by software. A transition from a LOW to HIGH initiates the interrupt at the end of the current instruction. This input is internally synchronized.
RESET	21	I	RESET: causes the processor to immediately terminate its present activity. The signal must transition LOW to HIGH and remain active HIGH for at least four clock cycles. It restarts execution, as described in the instruction set description, when RESET returns LOW. RESET is internally synchronized.
CLK	19	I	Clock: provides the basic timing for the processor and bus controller. It is asymmetric with a 33% duty cycle to provide optimized internal timing.
V _{CC}	40		V _{CC} : is the +5V ± 10% power supply pin.
GND	1, 20		GND: are the ground pins (Both pins must be connected to system ground).
MN/MX	33	I	Minimum/Maximum: indicates the mode in which the processor is to operate. The two modes are discussed in the following sections.

Pin Description (continued)

The following pin descriptions are for the 80C88 system in maximum mode (i.e., MN/MX = GND). Only the pin functions

which are unique to maximum mode are described; all other pin functions are as described above.

SYMBOL	PIN NUMBER	TYPE	NAME AND FUNCTION																																				
\bar{S}_2, \bar{S}_1 S_0	26-28	O	<p>Status: is active during clock high of T4, T1, and T2, and is returned to the passive state (1,1,1) during T3 or during Tw when READY is HIGH. This status is used by the 80C88 bus controller to generate all memory and I/O access control signals. Any change by \bar{S}_2, \bar{S}_1, or S_0 during T4 is used to indicate the beginning of a bus cycle, and the return to the passive state in T3 or Tw is used to indicate the end of a bus cycle.</p> <p>These signals are held internally to a high impedance logic one state during "grant sequence".</p>																																				
			<table border="1"> <thead> <tr> <th>\bar{S}_2</th><th>\bar{S}_1</th><th>S_0</th><th>CHARACTERISTICS</th></tr> </thead> <tbody> <tr> <td>0 (LOW)</td><td>0</td><td>0</td><td>Interrupt Acknowledge</td></tr> <tr> <td>0</td><td>0</td><td>1</td><td>Read I/O port</td></tr> <tr> <td>0</td><td>1</td><td>0</td><td>Write I/O port</td></tr> <tr> <td>0</td><td>1</td><td>1</td><td>Halt</td></tr> <tr> <td>1 (HIGH)</td><td>0</td><td>0</td><td>Code access</td></tr> <tr> <td>1</td><td>0</td><td>1</td><td>Read memory</td></tr> <tr> <td>1</td><td>1</td><td>0</td><td>Write memory</td></tr> <tr> <td>1</td><td>1</td><td>1</td><td>Passive</td></tr> </tbody> </table>	\bar{S}_2	\bar{S}_1	S_0	CHARACTERISTICS	0 (LOW)	0	0	Interrupt Acknowledge	0	0	1	Read I/O port	0	1	0	Write I/O port	0	1	1	Halt	1 (HIGH)	0	0	Code access	1	0	1	Read memory	1	1	0	Write memory	1	1	1	Passive
\bar{S}_2	\bar{S}_1	S_0	CHARACTERISTICS																																				
0 (LOW)	0	0	Interrupt Acknowledge																																				
0	0	1	Read I/O port																																				
0	1	0	Write I/O port																																				
0	1	1	Halt																																				
1 (HIGH)	0	0	Code access																																				
1	0	1	Read memory																																				
1	1	0	Write memory																																				
1	1	1	Passive																																				
$\bar{RQ}/\bar{GT}_0,$ \bar{RQ}/\bar{GT}_1	30, 31	I/O	<p>Request/Grant: pins are used by other local bus masters to force the processor to release the local bus at the end of the processor's current bus cycle. Each pin is bidirectional with \bar{RQ}/\bar{GT}_0 having higher priority than \bar{RQ}/\bar{GT}_1. \bar{RQ}/\bar{GT} has internal bus-hold high circuitry and, if unused, may be left unconnected. The request/grant sequence is as follows (see Figure 5):</p> <ol style="list-style-type: none"> 1. A pulse of one CLK wide from another local bus master indicates a local bus request ("hold") to the 80C88 (pulse 1). 2. During a T4 or T1 clock cycle, a pulse one clock wide from the 80C88 to the requesting master (pulse 2), indicates that the 80C88 has allowed the local bus to float and that it will enter the "grant sequence" state at the next CLK. The CPU's bus interface unit is disconnected logically from the local bus during "grant sequence". The same rules as for HOLD/HOLDA apply as for when the bus is released. 3. A pulse one CLK wide from the requesting master indicates to the 80C88 (pulse 3) that the "hold" request is about to end and that the 80C88 can reclaim the local bus at the next CLK. The CPU then enters T4. <p>Each master-master exchange of the local bus is a sequence of three pulses. There must be one idle CLK cycle after each bus exchange. Pulses are active LOW.</p> <p>If the request is made while the CPU is performing a memory cycle, it will release the local bus during T4 of the cycle when all the following conditions are met:</p> <ol style="list-style-type: none"> 1. Request occurs on or before T2. 2. Current cycle is not the low bit of a word. 3. Current cycle is not the first acknowledge of an interrupt acknowledge sequence. 4. A locked instruction is not currently executing. <p>If the local bus is idle when the request is made the two possible events will follow:</p> <ol style="list-style-type: none"> 1. Local bus will be released during the next clock. 2. A memory cycle will start within 3 clocks. Now the four rules for a currently active memory cycle apply with condition number 1 already satisfied. 																																				
LOCK	29	O	<p>LOCK: indicates that other system bus masters are not to gain control of the system bus while LOCK is active (LOW). The LOCK signal is activated by the "LOCK" prefix instruction and remains active until the completion of the next instruction. This signal is active LOW, and is held high internally during "grant sequence".</p>																																				
QS1, QS0	24, 25	O	<p>Queue Status: provide status to allow external tracking of the internal 80C88 instruction queue.</p> <p>The queue status is valid during the CLK cycle after which the queue operation is performed. Note that the queue status never goes to a high impedance state (floated).</p>																																				
--	34	O	<p>Pin 34 is always a logic one in the maximum mode and is internally held at a high impedance logic one during a "grant sequence".</p>																																				

Pin Description

The following pin function descriptions are for the 80C88 minimum mode (i.e., MN/MX = V_{cc}). Only the pin functions

which are unique to the minimum mode are described; all other pin functions are as described above.

SYMBOL	PIN NUMBER	TYPE	NAME AND FUNCTION
IO/M	28	O	Status Line: is an inverted maximum mode S ₂ . It is used to distinguish a memory access from an I/O access. IO/M becomes valid in the T4 preceding a bus cycle and remains valid until the final T4 of the cycle (I/O = HIGH, M = LOW). IO/M is held high impedance logic zero internally during local bus "hold acknowledge".
WR	29	O	Write strobe indicates that the processor is performing a write memory or write I/O cycle, depending on the state of the IO/M signal. WR is active for T ₂ , T ₃ , and T _w of any write cycle. It is active LOW, and is held to high impedance logic one internally during local bus "hold acknowledge".
INTA	24	O	INTA: is used as a read strobe for interrupt acknowledge cycles. It is active LOW during T ₂ , T ₃ , and T _w of each interrupt acknowledge cycle. Note that INTA is never floated.
ALE	25	O	Address Latch Enable: is provided by the processor to latch the address into the 82C82/82C83 address latch. It is a HIGH pulse active during clock low of T ₁ of any bus cycle. Note that ALE is never floated.
DT/R	27	O	Data Transmit/Receive: is needed in a minimum system that desires to use an 82C86/82C87 data bus transceiver. It is used to control the direction of data flow through the transceiver. Logically, DT/R is equivalent to S ₁ in the maximum mode, and its timing is the same as for IO/M (T = HIGH, R = LOW). This signal is held to a high impedance logic one internally during local bus "hold acknowledge".
DEN	26	O	Data Enable: is provided as an output enable for the 82C86/82C87 in a minimum system which uses the transceiver. DEN is active LOW during each memory and I/O access, and for INTA cycles. For a read or INTA cycle, it is active from the middle of T ₂ until the middle of T ₄ , while for a write cycle, it is active from the beginning of T ₂ until the middle of T ₄ . DEN is held to high impedance logic one internally during local bus "hold acknowledge".
HOLD, HLDA	30 31	I O	HOLD: indicates that another master is requesting a local bus "hold". To be acknowledged, HOLD must be active HIGH. The processor receiving the "hold" request will issue HLDA (HIGH) as an acknowledgment, in the middle of a T ₄ or T ₁ clock cycle. Simultaneous with the issuance of HLDA the processor will float the local bus and control lines. After HOLD is detected as being LOW, the processor lowers HLDA, and when the processor needs to run another cycle, it will again drive the local bus and control lines. Hold is not an asynchronous input. External synchronization should be provided if the system cannot otherwise guarantee the set up time.
SS0	34	O	Status line: is logically equivalent to S ₀ in the maximum mode. The combination of SS0, IO/M, and DT/R allows the system to completely decode the current bus cycle status. SS0 is held to high impedance logic one during local bus "hold acknowledge".

IO/M	DT/R	SS0	CHARACTERISTICS
1 (HIGH)	0	0	Interrupt Acknowledge
1	0	1	Read I/O port
1	1	0	Write I/O port
1	1	1	Halt
0 (LOW)	0	0	Code access
0	0	1	Read memory
0	1	0	Write memory
0	1	1	Passive

Functional Description

Static Operation

All 80C88 circuitry is static in design. Internal registers, counters and latches are static and require no refresh as with dynamic circuit design. This eliminates the minimum operating frequency restriction placed on other microprocessors. The CMOS 80C88 can operate from DC to the appropriate upper frequency limit of 5 MHz. The processor clock may be stopped in either state (high/low) and held there indefinitely. This type of operation is especially useful for system debug or power critical applications.

The 80C88 can be single stepped using only the CPU clock. This state can be maintained as long as is necessary. Single step clock operation allows simple interface circuitry to provide critical information for start-up.

Static design also allows very low frequency operation (as low as DC). In a power critical situation, this can provide extremely low power operation since 80C88 power dissipation is directly related to operating frequency. As the system frequency is reduced, so is the operating power until, at a DC input frequency, the power requirement is the 80C88 standby current.

Internal Architecture

The internal functions of the 80C88 processor are partitioned logically into two processing units. The first is the Bus Interface Unit (BIU) and the second is the Execution Unit (EU) as shown in the CPU block diagram.

These units can interact directly but for the most part perform as separate asynchronous operational processors. The bus interface unit provides the functions related to instruction fetching and queuing, operand fetch and store, and address relocation. This unit also provides the basic bus control. The overlap of instruction pre-fetching provided by this unit serves to increase processor performance through improved bus bandwidth utilization. Up to 4 bytes of the instruction stream can be queued while waiting for decoding and execution.

The instruction stream queuing mechanism allows the BIU to keep the memory utilized very efficiently. Whenever there is space for at least 1 byte in the queue, the BIU will attempt a byte fetch memory cycle. This greatly reduces "dead time" on the memory bus. The queue acts as a First-In-First-Out (FIFO) buffer, from which the EU extracts instruction bytes as required. If the queue is empty (following a branch instruction, for example), the first byte into the queue immediately becomes available to the EU.

The execution unit receives pre-fetched instructions from the BIU queue and provides un-relocated operand addresses to the BIU. Memory operands are passed through the BIU for processing by the EU, which passes results to the BIU for storage.

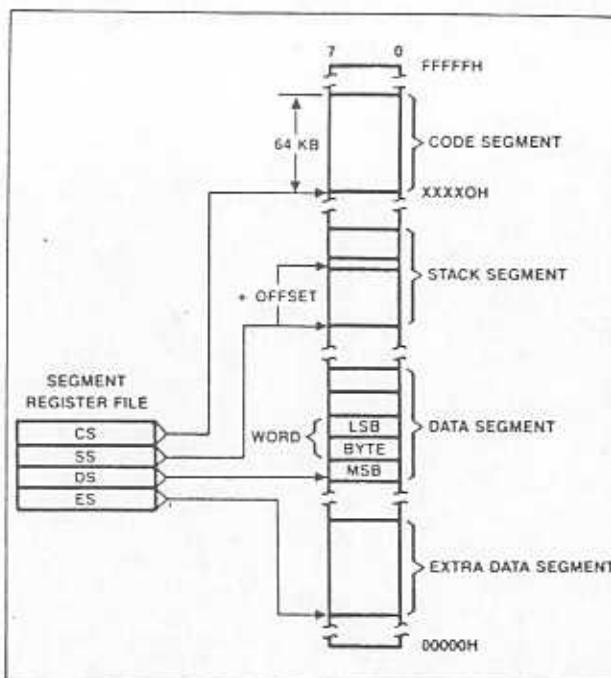


Figure 1. Memory Organization

Memory Organization

The processor provides a 20-bit address to memory which locates the byte being referenced. The memory is organized as a linear array of up to 1 million bytes, addressed as 00000(H) to FFFFF(H). The memory is logically divided into code, data, extra, and stack segments of up to 64K bytes each, with each segment falling on 16-byte boundaries. (See FIGURE 1).

All memory references are made relative to base addresses contained in high speed segment registers. The segment types were chosen based on the addressing needs of programs. The segment register to be selected is automatically chosen according to the rules of the following table. All information in one segment type share the same logical attributes (e.g., code or data). By structuring memory into relocatable areas of similar characteristics and by automatically selecting segment registers, programs are shorter, faster, and more structured.

Word (16-bit) operands can be located on even or odd address boundaries. For address and data operands, the least significant byte of the word is stored in the lower valued address location and the most significant byte in the next higher address location.

Memory Reference Need	Segment Register Used	Segment Selection Rule
Instructions	CODE (CS)	Automatic with all instruction prefetch
Stack	STACK (SS)	All stack pushes and pops. Memory references relative to BP base register except data references.
Local Data	DATA (DS)	Data references when: relative to stack, destination of string operation, or explicitly overridden.
External (Global) Data	EXTRA (ES)	Destination of string operations: Explicitly selected using a segment override.

Table 2.

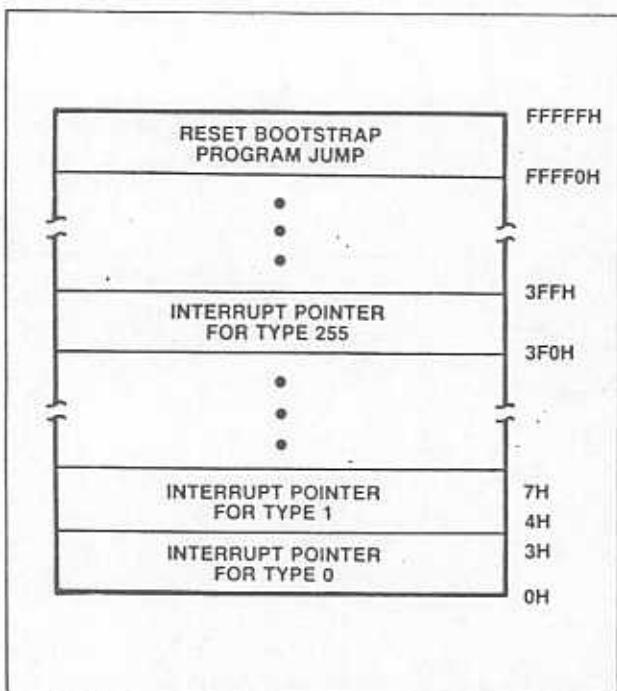


Figure 2. Reserved Memory Locations

The BIU will automatically execute two fetch or write cycles for 16-bit operands.

Certain locations in memory are reserved for specific CPU operations. (See FIGURE 2). Locations from addresses FFFF0H through FFFFFH are reserved for operations including a jump to the initial system initialization routine. Following RESET, the CPU will always begin execution at location FFFF0H where the jump must be located. Locations

00000H through 003FFH are reserved for interrupt operations. Four-byte pointers consisting of a 16-bit segment address and a 16-bit offset address direct program flow to one of the 256 possible interrupt service routines. The pointer elements are assumed to have been stored at their respective places in reserved memory prior to the occurrence of interrupts.

Minimum and Maximum Modes

The requirements for supporting minimum and maximum 80C88 systems are sufficiently different that they cannot be done efficiently with 40 uniquely defined pins. Consequently, the 80C88 is equipped with a strap pin (MN/MX) which defines the system configuration. The definition of a certain subset of the pins changes, dependent on the condition of the strap pin. When the MN/MX pin is strapped to GND, the 80C88 defines pins 24 through 31 and 34 in maximum mode. When the MN/MX pin is strapped to VCC, the 80C88 generates bus control signals itself on pins 24 through 31 and 34.

The minimum mode 80C88 can be used with either a multiplexed or demultiplexed bus. This architecture provides the 80C88 processing power in a highly integrated form.

The demultiplexed mode requires one latch (for 64K addressability) or two latches (for a full megabyte of addressing). A third latch can be used for buffering if the address bus loading requires it. An 82C86 or 82C87 transceiver can also be used if data bus buffering is required. (See FIGURE 3.) The 80C88 provides DEN and DT/R to control the transceiver, and ALE to latch the addresses. This configuration of the minimum mode provides the standard demultiplexed bus structure with heavy bus buffering and relaxed bus timing requirements.

The maximum mode employs the 82C88 bus controller (See FIGURE 4). The 82C88 decodes status lines S0, S1, and S2, and provides the system with all bus control signals. Moving the bus control to the 82C88 provides better source and sink current capability to the control lines, and frees the 80C88 pins for extended large system features. Hardware lock, queue status, and two request/grant interfaces are provided by the 80C88 in maximum mode. These features allow co-processors in local bus and remote bus configurations.

80C88

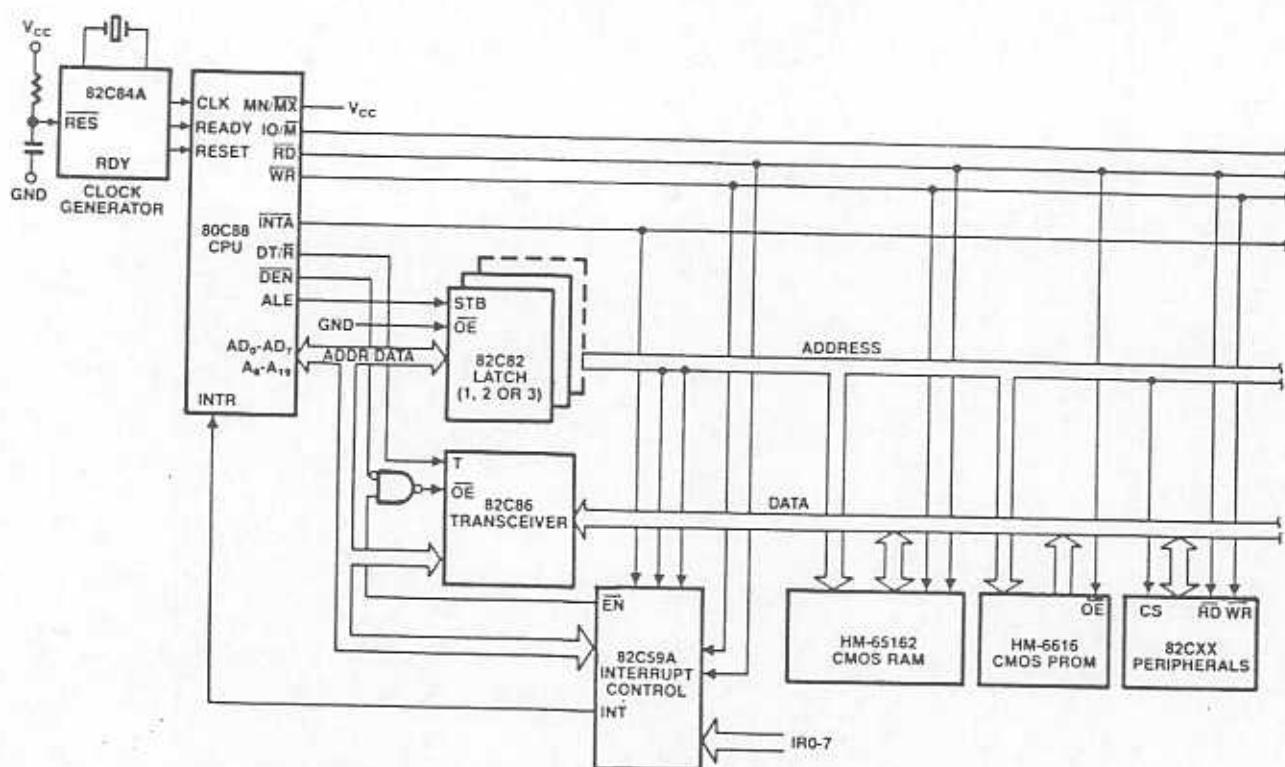


Figure 3. Demultiplexed Bus Configuration

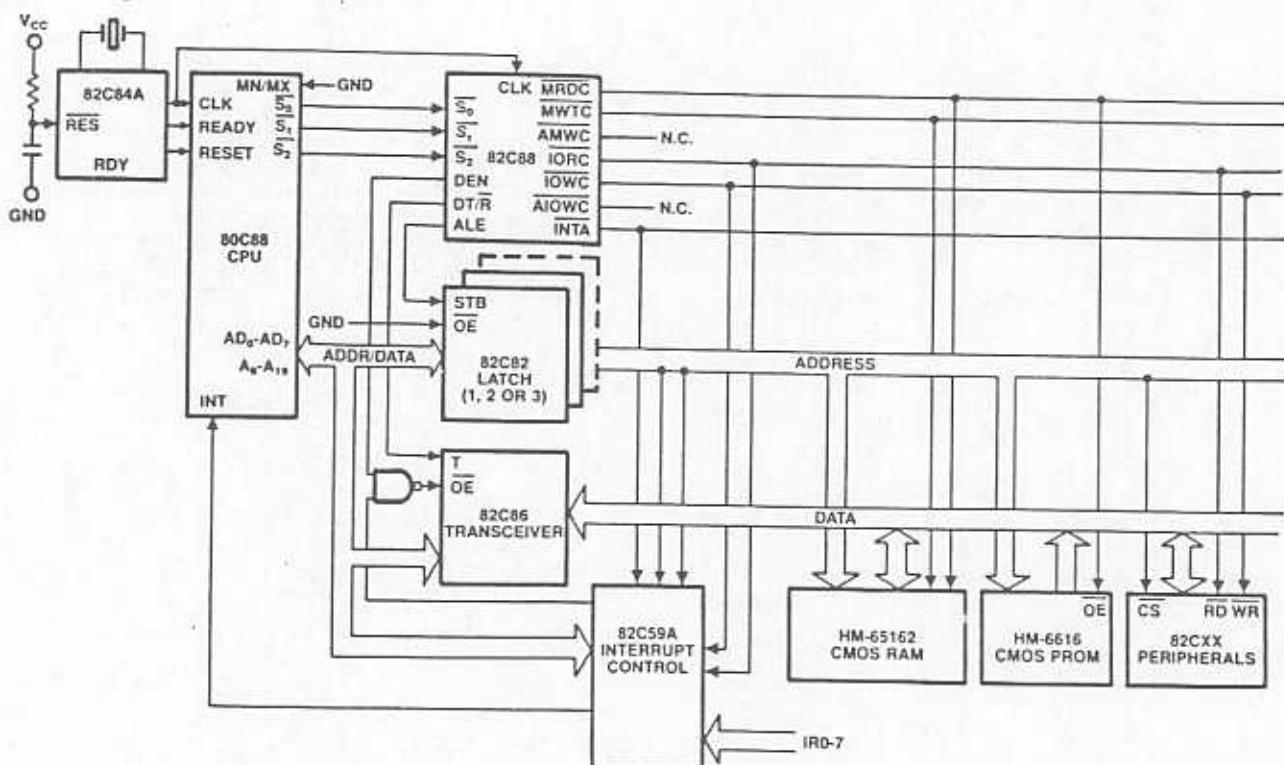


Figure 4. Fully Buffered System Using Bus Controller

Bus Operation

The 80C88 address/data bus is broken into three parts – the lower eight address/data bits (AD0-AD7), the middle eight address bits (A8-A15), and the upper four address bits (A16-A19). The address/data bits and the highest four address bits are time multiplexed. This technique provides the most efficient use of pins on the processor, permitting the use of a standard 40 lead package. The middle eight address bits are not multiplexed, i.e. they remain valid throughout each bus cycle. In addition, the bus can be demultiplexed at the processor with a single address latch if a standard, non-multiplexed bus is desired for the system.

Each processor bus cycle consists of at least four CLK cycles. These are referred to as T1, T2, T3, and T4. (See FIGURE 5). The address is emitted from the processor during T1 and data transfer occurs on the bus during T3 and T4. T2 is used primarily for changing the direction of the bus during read operations. In the event that a "NOT READY" indication is given by the addressed device, "wait" states (T_w) are inserted between T3 and T4. Each inserted "wait" state is of the same duration as a CLK cycle. Periods can occur between 80C88 driven bus cycles. These are referred to as "idle" states (T_i), or inactive CLK cycles. The processor uses these cycles for internal housekeeping.

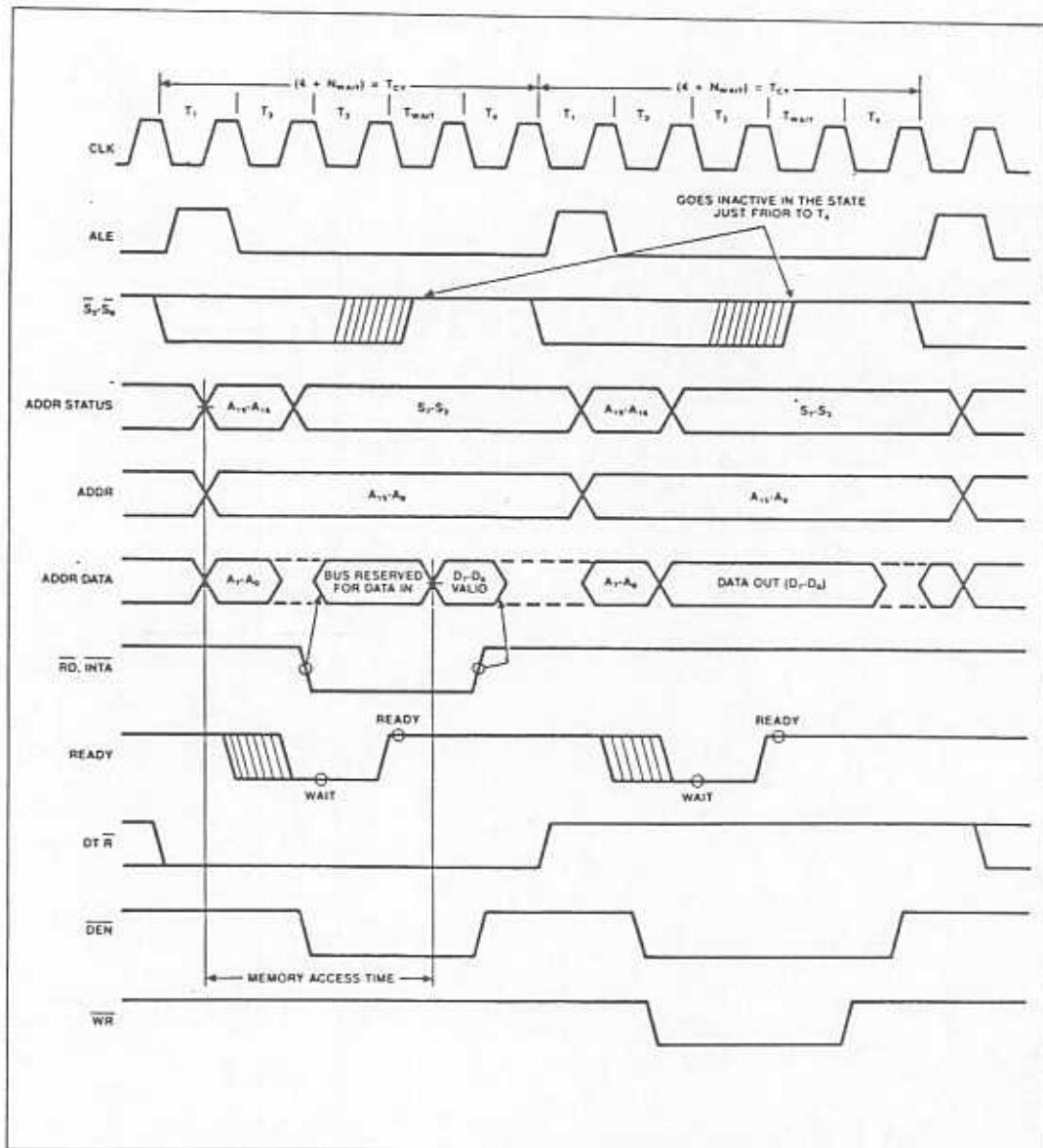


Figure 5. Basic System Timing

During T1 of any bus cycle, the ALE (address latch enable) signal is emitted (by either the processor or the 82C88 bus controller, depending on the MN/MX strap). At the trailing edge of this pulse, a valid address and certain status information for the cycle may be latched.

Status bits $\overline{S_0}$, $\overline{S_1}$, and $\overline{S_2}$ are used by the bus controller, in maximum mode, to identify the type of bus transaction according to the following table:

$\overline{S_2}$	$\overline{S_1}$	$\overline{S_0}$	CHARACTERISTICS
0 (LOW)	0	0	Interrupt Acknowledge
0	0	1	Read I/O
0	1	0	Write I/O
0	1	1	Halt
1 (HIGH)	0	0	Instruction Fetch
1	0	1	Read Data from Memory
1	1	0	Write Data to Memory
1	1	1	Passive (no bus cycle)

Table 3.

Status bits S_3 through S_6 are multiplexed with high order address bits and are therefore valid during T2 through T4. S_3 and S_4 indicate which segment register was used for this bus cycle in forming the address according to the following table:

S_4	S_3	CHARACTERISTICS
0 (LOW)	0	Alternate Data (extra segment)
0	1	Stack
1 (HIGH)	0	Code or None
1	1	Data

Table 4.

S_5 is a reflection of the PSW interrupt enable bit. S_6 is always equal to 0.

I/O Addressing

In the 80C88, I/O operations can address up to a maximum of 64K I/O registers. The I/O address appears in the same format as the memory address on bus lines A15-A0. The address lines A19-A16 are zero in I/O operations. The variable I/O instructions, which use register DX as a pointer, have full address capability, while the direct I/O instructions directly address one or two of the 256 I/O byte locations in page 0 of the I/O address space. I/O ports are addressed in the same manner as memory locations.

Designers familiar with the 8085 or upgrading an 8085 design should note that the 8085 address I/O with an 8-bit address on both halves of the 16-bit address bus. The 80C88 uses a full 16-bit address on its lower 16 address lines.

External Interface

Processor Reset and Initialization

Processor initialization or start up is accomplished with activation (HIGH) of the RESET pin. The 80C88 RESET is required to be HIGH for greater than four clock cycles. The 80C88 will terminate operations on the high-going edge of RESET and will remain dormant as long as RESET is HIGH. The low-going transition of RESET triggers an internal reset sequence for approximately 7 clock cycles. After this interval the 80C88 operates normally, beginning with the instruction in absolute location FFFF0H (see FIGURE 2). The RESET input is internally synchronized to the processor clock. At initialization, the HIGH to LOW transition of RESET must occur no sooner than 50 μ s after power up, to allow complete initialization of the 80C88.

If INTR is asserted sooner than nine clock cycles after the end of RESET, the processor may execute one instruction before responding to the interrupt.

Bus Hold Circuitry

To avoid high current conditions caused by floating inputs to CMOS devices and to eliminate the need for pull-up/down resistors, "bus-hold" circuitry has been used on 80C88 pins 2-16, 26-32 and 34-39 (see FIGURE 6A, 6B). These circuits maintain a valid logic state if no driving source is present (i.e.,

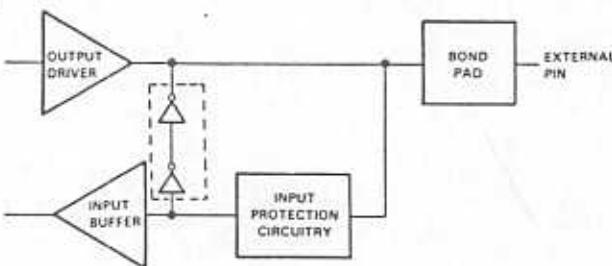


Figure 6A. Bus hold circuitry pin 2-16, 35-39.

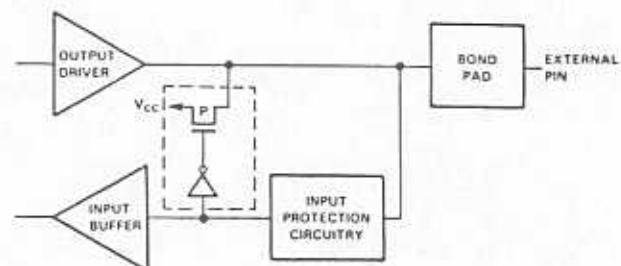


Figure 6B. Bus hold circuitry pin 26-32, 34.

an unconnected pin or a driving source which goes to a high impedance state).

To overdrive the "bus hold" circuits, an external driver must be capable of supplying $400\mu A$ minimum sink or source current at valid input voltage levels. Since this "bus hold" circuitry is active and not a "resistive" type element, the associated power supply current is negligible. Power dissipation is significantly reduced when compared to the use of passive pull-up resistors.

Interrupt Operations

Interrupt operations fall into two classes: software or hardware initiated. The software initiated interrupts and software aspects of hardware interrupts are specified in the instruction set description. Hardware interrupts can be classified as non-maskable or maskable.

Interrupts result in a transfer of control to a new program location. A 256 element table containing address pointers to the interrupt service program locations resides in absolute locations 0 through 3FFH (see FIGURE 2), which are reserved for this purpose. Each element in the table is 4 bytes in size and corresponds to an interrupt "type." An interrupting device supplies an 8-bit type number, during the interrupt acknowledge sequence, which is used to vector through the appropriate element to the new interrupt service program location.

Non-Maskable Interrupt (NMI)

The processor provides a single non-maskable interrupt (NMI) pin which has higher priority than the maskable interrupt request (INTR) pin. A typical use would be to activate a power failure routine. The NMI is edge-triggered on a LOW to HIGH transition. The activation of this pin causes a type 2 interrupt.

NMI is required to have a duration in the HIGH state of greater than two clock cycles, but is not required to be synchronized to the clock. Any high going transition of NMI is latched on-chip and will be serviced at the end of the current instruction or between whole moves (2 bytes in the case of word moves) of a block type instruction. Worst case response to NMI would be for multiply, divide, and variable shift instructions. There is no specification on the occurrence of the low-going edge; it may

occur before, during, or after the servicing of NMI. Another high-going edge triggers another response if it occurs after the start of the NMI procedure.

The signal must be free of logical spikes in general and be free of bounces on the low-going edge to avoid triggering extraneous responses.

Maskable Interrupt (INTR)

The 80C88 provides a single interrupt request input (INTR) which can be masked internally by software with the resetting of the interrupt enable (IF) flag bit. The interrupt request signal is level triggered. It is internally synchronized during each clock cycle on the high-going edge of CLK.

To be responded to, INTR must be present (HIGH) during the clock period preceding the end of the current instruction or the end of a whole move for a block type instruction. INTR may be removed anytime after the falling edge of the first INTA signal. During interrupt response sequence, further interrupts are disabled. The enable bit is reset as part of the response to any interrupt (INTR, NMI, software interrupt, or single step). The FLAGS register, which is automatically pushed onto the stack, reflects the state of the processor prior to the interrupt. The enable bit will be zero until the old FLAGS register is restored, unless specifically set by an instruction.

During the response sequence (see FIGURE 7), the processor executes two successive (back to back) interrupt acknowledge cycles. The 80C88 emits the LOCK signal (maximum mode only) from T2 of the first bus cycle until T2 of the second. A local bus "hold" request will not be honored until the end of the second bus cycle. In the second bus cycle, a byte is fetched from the external interrupt system (e.g., 82C59A PIC) which identifies the source (type) of the interrupt. This byte is multiplied by four and used as a pointer into the interrupt vector lookup table.

An INTR signal left HIGH will be continually responded to within the limitations of the enable bit and sample period. INTR may be removed anytime after the falling edge of the first INTA signal. The interrupt return instruction includes a flags pop which returns the status of the original interrupt enable bit when it restores the flags.

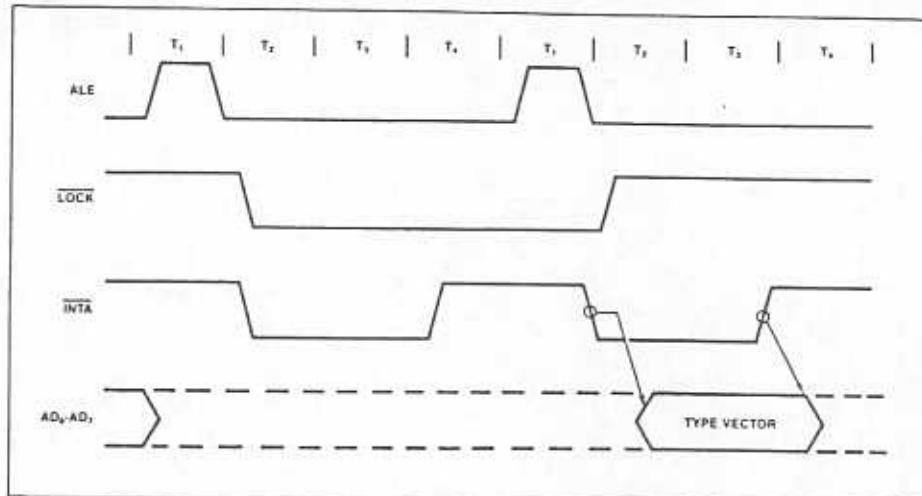


Figure 7. Interrupt Acknowledge Sequence

Halt

When a software HALT instruction is executed, the processor indicates that it is entering the HALT state in one of two ways, depending upon which mode is strapped. In minimum mode, the processor issues ALE, delayed by one clock cycle, to allow the system to latch the halt status. Halt status is available on IO/M, DT/R, and S₀. In maximum mode, the processor issues appropriate HALT status on S₂, S₁, and S₀, and the 80C88 bus controller issues one ALE. The 80C88 will not leave the HALT state when a local bus hold is entered while in HALT. In this case, the processor reissues the HALT indicator at the end of the local bus hold. An interrupt request or RESET will force the 80C88 out of the HALT state.

Read/Modify/Write (Semaphore) Operations Via LOCK

The LOCK status information is provided by the processor when consecutive bus cycles are required during the execution of an instruction. This allows the processor to perform read/modify/write operations on memory (via the "exchange register with memory" instruction), without another system bus master receiving intervening memory cycles. This is useful in multiprocessor system configurations to accomplish "test and set lock" operations. The LOCK signal is activated (LOW) in the clock cycle following decoding of the LOCK prefix instruction. It is deactivated at the end of the last bus cycle of the instruction following the LOCK prefix. While LOCK is active, a request on a RQ/GT pin will be recorded, and then honored at the end of the LOCK.

External Synchronization Via TEST

As an alternative to interrupts, the 80C88 provides a single software-testable input pin (TEST). This input is utilized by executing a WAIT instruction. The single WAIT instruction is repeatedly executed until the TEST input goes active (LOW). The execution of WAIT does not consume bus cycles once the queue is full.

If a local bus request occurs during WAIT execution, the 80C88 3-states all output drivers while inputs and I/O pins are held at valid logic levels by internal bus-hold circuits. If interrupts are enabled, the 80C88 will recognize interrupts and process them. The WAIT instruction is then refetched, and reexecuted.

Basic System Timing

In minimum mode, the MN/MX pin is strapped to V_{CC} and the processor emits bus control signals (RD, WR, IO/M, etc.) directly. In maximum mode, the MN/MX pin is strapped to GND and the processor emits coded status information which the 80C88 bus controller uses to generate MULTIBUS™ compatible bus control signals.

System Timing – Minimum System

The read cycle begins in T1 with the assertion of the address latch enable (ALE) signal (See FIGURE 5). The trailing (low going) edge of this signal is used to latch the address information, which is valid on the address/data bus (AD0-AD7) at this time, into the 82C82/82C83 latch. Address lines A8 through A15 do not need to be latched because they remain valid throughout the bus cycle. From T1 to T4 the IO/M signal indicates a memory or I/O operation. At T2 the address is removed from the address/data bus and the bus is held at the last valid logic state by internal bus-hold devices. The read control signal is also asserted at T2. The read (RD) signal causes the addressed device to enable its data bus drivers to

the local bus. Some time later, valid data will be available on the bus and the addressed device will drive the READY line HIGH. When the processor returns the read signal to a HIGH level, the addressed device will again 3-state its bus drivers. If a transceiver (82C86/82C87) is required to buffer the local bus, signals DT/R and DEN are provided by the 80C88.

A write cycle also begins with the assertion of ALE and the emission of the address. The IO/M signal is again asserted to indicate a memory or I/O write operation. In T2, immediately following the address emission, the processor emits the data to be written into the addressed location. This data remains valid until at least the middle of T4. During T2, T3, and Tw, the processor asserts the write control signal. The write (WR) signal becomes active at the beginning of T2, as opposed to the read, which is delayed somewhat into T2 to provide time for output drivers to become inactive.

The basic difference between the interrupt acknowledge cycle and a read cycle is that the interrupt acknowledge (INTA) signal is asserted in place of the read (RD) signal and the address bus is held at the last valid logic state by internal bus-hold devices (see FIGURE 6). In the second of two successive INTA cycles, a byte of information is read from the data bus, as supplied by the interrupt system logic (i.e., 82C59A priority interrupt controller). This byte identifies the source (type) of the interrupt. It is multiplied by four and used as a pointer into the interrupt vector lookup table, as described earlier.

Bus Timing – Medium Complexity Systems

For medium complexity systems, the MN/MX pin is connected to GND and the 80C88 bus controller is added to the system, as well as an 82C82/82C83 latch for latching the system address, and an 82C86/82C87 transceiver to allow for bus loading greater than the 80C88 is capable of handling (see FIGURE 8). Signals ALE, DEN, and DT/R are generated by the 80C88 instead of the processor in this configuration, although their timing remains relatively the same. The 80C88 status outputs (S₂, S₁, and S₀) provide type of cycle information and become 80C88 inputs. This bus cycle information specifies read (code, data, or I/O), write (data or I/O), interrupt acknowledge, or software halt. The 80C88 thus issues control signals specifying memory read or write, I/O read or write, or interrupt acknowledge. The 80C88 provides two types of write strobes, normal and advanced, to be applied as required. The normal write strobes have data valid at the leading edge of write. The advanced write strobes have the same timing as read strobes, and hence, data is not valid at the leading edge of write. The 82C86/82C87 transceiver receives the usual T and OE inputs from the 80C88 DT/R and DEN outputs.

The pointer into the interrupt vector table, which is passed during the second INTA cycle, can derive from an 82C59A located on either the local bus or the system bus. If the master 82C59A priority interrupt controller is positioned on the local bus, the 82C86/82C87 transceiver must be disabled when reading from the master 82C59A during the interrupt acknowledge sequence and software "poll".

The 80C88 Compared To The 80C86

The 80C88 CPU is an 8-bit processor designed around the 8086 internal structure. Most internal functions of the 80C88 are identical to the equivalent 80C86 functions. The 80C88 handles the external bus the same way the 80C86 does with the distinction of handling only 8 bits at a time. Sixteen-bit

operands are fetched or written in two consecutive bus cycles. Both processors will appear identical to the software engineer, with the exception of execution time. The internal register structure is identical and all instructions have the same end result. Internally, there are three differences between the 80C88 and the 80C86. All changes are related to the 8-bit bus interface.

- The queue length is 4 bytes in the 80C88, whereas the 80C86 queue contains 6 bytes, or three words. The queue was shortened to prevent overuse of the bus by the BIU when prefetching instructions. This was required because of the additional time necessary to fetch instructions 8 bits at a time.
- To further optimize the queue, the prefetching algorithm was changed. The 80C88 BIU will fetch a new instruction to load into the queue each time there is a 1 byte space available in the queue. The 80C86 waits until a 2-byte space is available.
- The internal execution time of the instruction set is affected by the 8-bit interface. All 16-bit fetches and writes from/to memory take an additional four clock cycles. The CPU is also limited by the speed of instruction fetches. This latter problem only occurs when a series of simple operations occur. When the more sophisticated instructions of the 80C88 are being used, the queue has time to fill and the execution proceeds as fast as the execution unit will allow.

The 80C88 and 80C86 are completely software compatible by virtue of their identical execution units. Software that is system dependent may not be completely transferable, but software that is not system dependent will operate equally as well on an 80C88 or an 80C86.

The hardware interface of the 80C88 contains the major differences between the two CPUs. The pin assignments are nearly identical, however, with the following functional changes:

- A8-A15 – These pins are only address outputs on the 80C88. These address lines are latched internally and remain valid throughout a bus cycle in a manner similar to the 8085 upper address lines.
- BHE has no meaning on the 80C88 and has been eliminated.
- SS0 provides the S0 status information in the minimum mode. This output occurs on pin 34 in minimum mode only. DT/R, IO/M, and SS0 provide the complete bus status in minimum mode.
- IO/M has been inverted to be compatible with the 8085 bus structure.
- ALE is delayed by one clock cycle in the minimum mode when entering HALT, to allow the status to be latched with ALE.

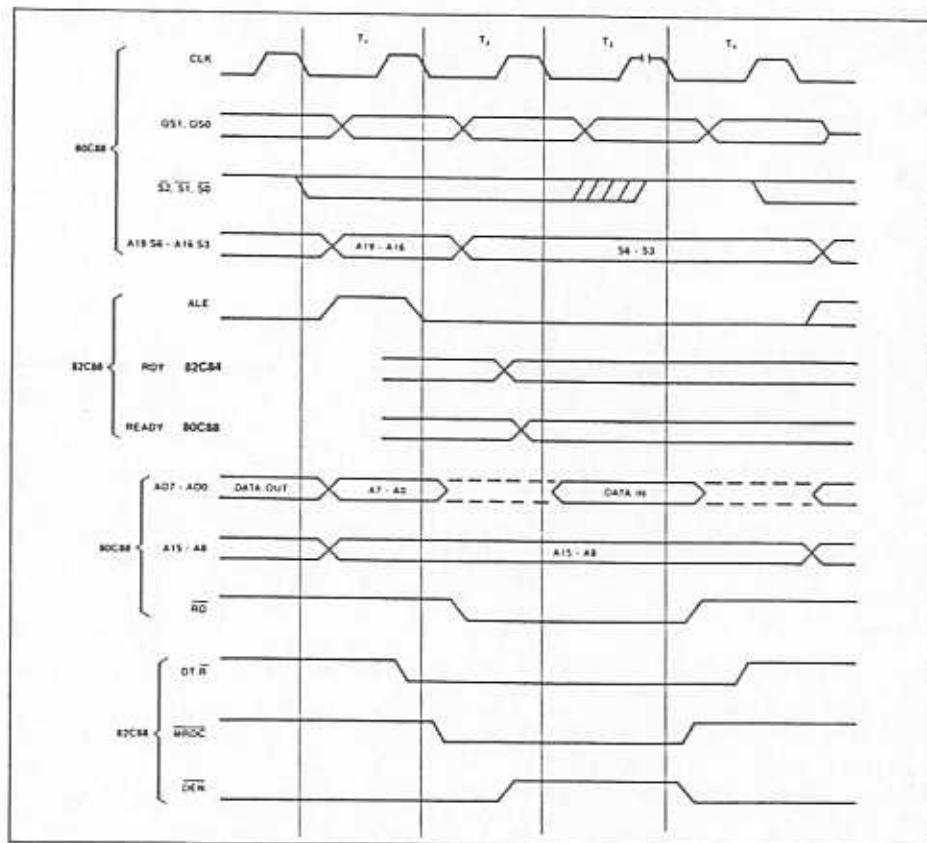


Figure 8. Medium Complexity System Timing

*Specifications***ABSOLUTE MAXIMUM RATINGS**

Supply Voltage	+8.0 Volts	Operating Temperature Range	
Operating Voltage Range	+4V to +7V	Commercial	0°C to +70°C
Input Voltage Applied	GND - 2.0V to 6.5V	Industrial	-40°C to +85°C
Output or I/O Voltage Applied	GND - 0.5V to VCC + 0.5V	Military	-55°C to +125°C
Storage Temperature Range	-65°C to +150°C	Maximum Package Power Dissipation	1 Watt

CAUTION: Stresses above those listed in the "ABSOLUTE MAXIMUM RATINGS" may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

D.C. ELECTRICAL CHARACTERISTICS

VCC = 5.0V ± 10%; TA = 0°C to +70°C (C80C88); TA = -40°C to +85°C (I80C88); TA = -55°C to +125°C (M80C88)

SYMBOL	PARAMETER	MIN	MAX	UNITS	TEST CONDITIONS
VIH	Logical One Input Voltage	2.0		V	C80C88, I80C88 M80C88
VIL	Logical Zero Input Voltage		0.8	V	
VIHC	CLK Logical One Input Voltage	VCC - 0.8V		V	
VILC	CLK Logical Zero Input Voltage		0.8	V	
VOH	Output High Voltage	3.0		V	IOH = -2.5mA
		VCC - 0.4		V	IOH = -100µA
VOL	Output Low Voltage		0.4	V	IOL = +2.5mA
IIL	Input Leakage Current	-1.0	1.0	µA	OV ≤ VIN ≤ VCC
IBHH	Input Current - Bus Hold High	-40	-400	µA	VIN = 3.0V (see Note 1)
IBHL	Input Current - Bus Hold Low	40	400	µA	VIN = 0.8V (see Note 2)
IO	Output Leakage Current	-10.0	10.0	µA	OV ≤ VO ≤ VCC
ICCSB	Standby Power Supply Current		500	µA	VCC = 5.5V See Note 3.
ICCOP	Operating Power Supply Current		10	mA/MHz	Vcc = 5.5V

CAPACITANCE

TA = 25°C; VCC = GND = OV; VIN = +5V or GND

SYMBOL	PARAMETER	MIN	MAX	UNITS	TEST CONDITIONS
CIN*	Input Capacitance		5	pf	FREQ = 1MHz Unmeasured pins returned to GND
COUT*	Output Capacitance		15	pf	
CI/O*	I/O Capacitance		20	pf	

* Guaranteed and sampled, but not 100% tested

Note 1: IBHH should be measured after raising VIN to VCC and then lowering to 3.0V on the following pins: 2-16, 26-32, 34-39.

Note 2: IBHL should be measured after lowering VIN to GND and then raising to 0.8V on the following pins: 2-16, 35-39.

Note 3: ICCSB tested during clock high time after HALT instruction execution. VIN = VCC or GND VCC = 5.5V Outputs unloaded

*Specifications (continued)***A.C. ELECTRICAL CHARACTERISTICS**

V_{CC} = 5V ± 10%; (C80C88: T_A = 0°C to +70°C)
 (I80C88: T_A = -40°C to +85°C)
 (M80C88: T_A = -55°C to +125°C)

MINIMUM COMPLEXITY SYSTEM TIMING REQUIREMENTS

SYMBOL	PARAMETER	80C88-4		80C88		UNITS	TEST CONDITIONS
		MIN	MAX	MIN	MAX		
TCLCL	CLK Cycle Period	250		200		ns	
TCLCH	CLK Low Time	151		118		ns	
TCHCL	CLK High Time	85		69		ns	
TCH1CH2	CLK Rise Time		10		10	ns	From 1.0V to 3.5V
TCL2CL1	CLK Fall Time		10		10	ns	From 3.5V to 1.0V
TDVCL	Data in Setup Time	30		30		ns	
TCLDX	Data in Hold Time	10		10		ns	
TR1VCL	RDY Setup Time into 82C84A (See Notes 1, 2)	35		35		ns	
TCLR1X	RDY Hold Time into 82C84A (See Notes 1, 2)	0		0		ns	
TRYHCH	READY Setup Time into 80C88	118		118		ns	
TCHRYX	READY Hold Time into 80C88	30		30		ns	
TRYLCL	READY Inactive to CLK (See Note 3)	-8		-8		ns	
THVCH	HOLD Setup Time	35		35		ns	
TINVCH	INTR, NMI, TEST Setup Time (See Note 2)	30		30		ns	
TILIH	Input Rise Time (Except CLK)		15		15	ns	From 0.8V to 2.0V
TIHIL	Input Fall Time (Except CLK)		15		15	ns	From 2.0V to 0.8V

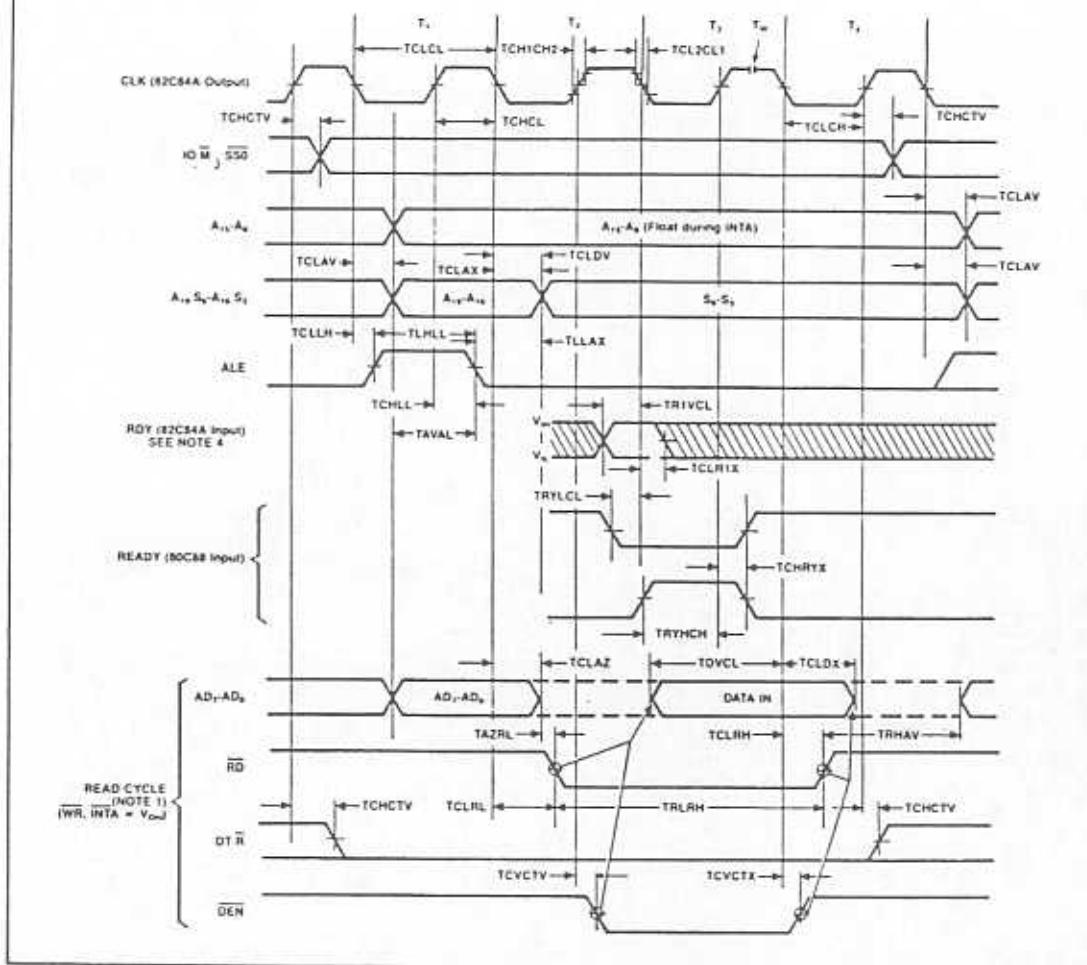
A.C. ELECTRICAL CHARACTERISTICS (Continued)
MINIMUM COMPLEXITY SYSTEM TIMING RESPONSES

SYMBOL	PARAMETER	80C88-4		80C88		UNITS	TEST CONDITIONS
		MIN	MAX	MIN	MAX		
TCLAV	Address Valid Delay	10	110	10	110	ns	
TCLAX	Address Hold Time	10		10		ns	
TCLAZ	Address Float Delay	TCLAX	80	TCLAX	80	ns	
TCHSZ	Status Float Delay		80		80	ns	
TLHLL	ALE Width	TCLCH-20		TCLCH-20		ns	
TCLLH	ALE Active Delay		80		80	ns	
TCHLL	ALE Inactive Delay		85		85	ns	
TLLAX	Address Hold Time to ALE Inactive	TCHCL-10		TCHCL-10		ns	
TCLDV	Data Valid Delay	10	110	10	110	ns	$C_L = 100 \text{ pF}$ for all 80C88 Outputs in addition to internal loads
TCHDX	Data Hold Time	10		10		ns	
TWHDX	Data Hold Time After WR	TCLCH-30		TCLCH-30		ns	
TCVCTV	Control Active Delay 1	10	110	10	110	ns	
TCHCTV	Control Active Delay 2	10	110	10	110	ns	
TCVCTX	Control Inactive Delay	10	110	10	110	ns	
TAZRL	Address Float to READ Active	0		0		ns	
TCLRL	RD Active Delay	10	165	10	165	ns	
TCLRH	RD Inactive Delay	10	150	10	150	ns	
TRHAV	RD Inactive to Next Address Active	TCLCL-45		TCLCL-45		ns	
TCLHAV	HILDA Valid Delay	10	160	10	160	ns	
TRLRH	RD Width	2TCLCL-75		2TCLCL-75		ns	
TWLWH	WR Width	2TCLCL-60		2TCLCL-60		ns	
TAVAL	Address Valid to ALE Low	TCLCH-60		TCLCH-60		ns	
TOLOH	Output Rise Time		15		15	ns	From 0.8V to 2.0V
TOHOL	Output Fall Time		15		15	ns	From 2.0V to 0.8V

NOTES: 1. Signal at 82C84A shown for reference only.
 2. Setup requirement for asynchronous signal only to guarantee recognition at next clock.
 3. Applies only to T2 state (8 nanoseconds into T3).

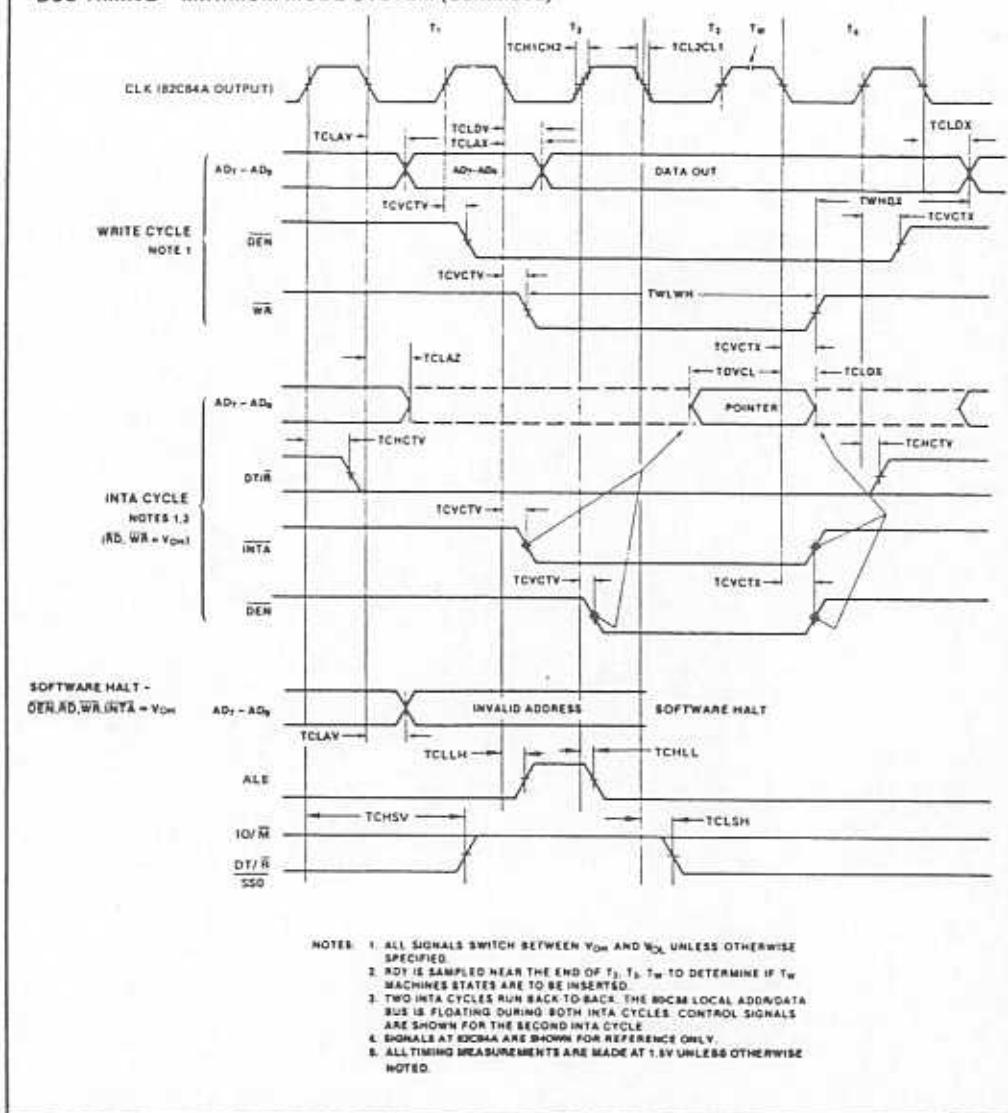
Waveforms

BUS TIMING – MINIMUM MODE SYSTEM



Waveforms (Continued)

BUS TIMING—MINIMUM MODE SYSTEM (Continued)



A.C. ELECTRICAL CHARACTERISTICS (Continued)
MAX MODE SYSTEM (USING 82C88 BUS CONTROLLER) TIMING REQUIREMENTS

SYMBOL	PARAMETER	80C88-4		80C88		UNITS	TEST CONDITIONS
		MIN	MAX	MIN	MAX		
TCLCL	CLK Cycle Period	250		200		ns	
TCLCH	CLK Low Time	151		118		ns	
TCHCL	CLK High Time	85		69		ns	
TCH1CH2	CLK Rise Time		10		10	ns	
TCL2CL1	CLK Fall Time		10		10	ns	From 1.0V to 3.5V
TDVCL	Data in Setup Time	30		30		ns	From 3.5V to 1.0V
TCLDX	Data in Hold Time	10		10		ns	
TR1VCL	RDY Setup Time into 82C84 (See Notes 1, 2)	35		35		ns	
TCLR1X	RDY Hold Time into 82C84 (See Notes 1, 2)	0		0		ns	
TRYHCH	READY Setup Time into 80C88	118		118		ns	
TCHRYX	READY Hold Time into 80C88	30		30		ns	
TRYLCL	READY Inactive to CLK (See Note 3)	-8		-8		ns	
TINVCH	Setup Time for Recognition (INTR, NMI, TEST) (See Note 2)	30		30		ns	
TGVCH	\bar{RQ}/\bar{GT} Setup Time	30		30		ns	
TCHGX	\bar{RQ} Hold Time into 80C88 (See Note 4)	40	TCHCL	40	TCHCL	ns	
TI1H	Input Rise Time (Except CLK)		15		15	ns	From 0.8V to 2.0V
TI1L	Input Fall Time (Except CLK)		15		15	ns	From 2.0V to 0.8V

A.C. CHARACTERISTICS

MAX MODE SYSTEM (USING 82C88 BUS CONTROLLER) TIMING RESPONSES

SYMBOL	PARAMETER	80C88-4		80C88		UNITS	TEST CONDITIONS
		MIN	MAX	MIN	MAX		
TCLML	Command Active Delay (See Note 1)	5	35	5	35	ns	
TCLMH	Command Inactive Delay (See Note 1)	5	35	5	35	ns	
TRYHSH	READY Active to Status Passive (See Notes 3, 5)		110		110	ns	
TCHSV	Status Active Delay	10	110	10	110	ns	
TCLSH	Status Inactive Delay (See Note 5)	10	130	10	130	ns	
TCLAV	Address Valid Delay	10	110	10	110	ns	
TCLAX	Address Hold Time	10		10		ns	
TCLAZ	Address Float Delay	TCLAX	80	TCLAX	80	ns	
TCHSZ	Status Float Delay		80		80	ns	
TSVLH	Status Valid to ALE High (See Note 1)		20		20	ns	
TSVMCH	Status Valid to MCE High (See Note 1)		30		30	ns	
TCLLH	CLK Low to ALE Valid (See Note 1)		15		15	ns	
TCLMCH	CLK Low to MCE High (See Note 1)		25		25	ns	
TCHLL	ALE Inactive Delay (See Note 1)	4	18	4	18	ns	
TCLMCL	MCE Inactive Delay (See Note 1)		15		15	ns	$C_L = 100 \text{ pF}$ for all 80C88 Outputs in addition to internal loads
TCLDV	Data Valid Delay	10	110	10	110	ns	
TCHDX	Data Hold Time	10		10		ns	
TCVN	Control Active Delay (See Note 1)	5	45	5	45	ns	
TCVNX	Control Inactive Delay (See Note 1)	10	45	10	45	ns	
TAZRL	Address Float to Read Active	0		0		ns	
TCLRL	RD Active Delay	10	165	10	165	ns	
TCLRH	RD Inactive Delay	10	150	10	150	ns	
TRHAV	RD Inactive to Next Address Active	TCLCL-45		TCLCL-45		ns	
TCHDTL	Direction Control Active Delay (See Note 1)		50		50	ns	
TCHDTH	Direction Control Inactive Delay (See Note 1)		30		30	ns	
TCLGL	GT Active Delay		85		85	ns	
TCLGH	GT Inactive Delay		85		85	ns	
TRLRH	RD Width	2TCLCL-75				ns	
TOLOH	Output Rise Time		15		15	ns	From 0.8V to 2.0V
TOHOL	Output Fall Time		15		15	ns	

NOTES: 1. Signal at 82C84A shown for reference only.

2. Setup requirement for asynchronous signal only to guarantee recognition at next clock.

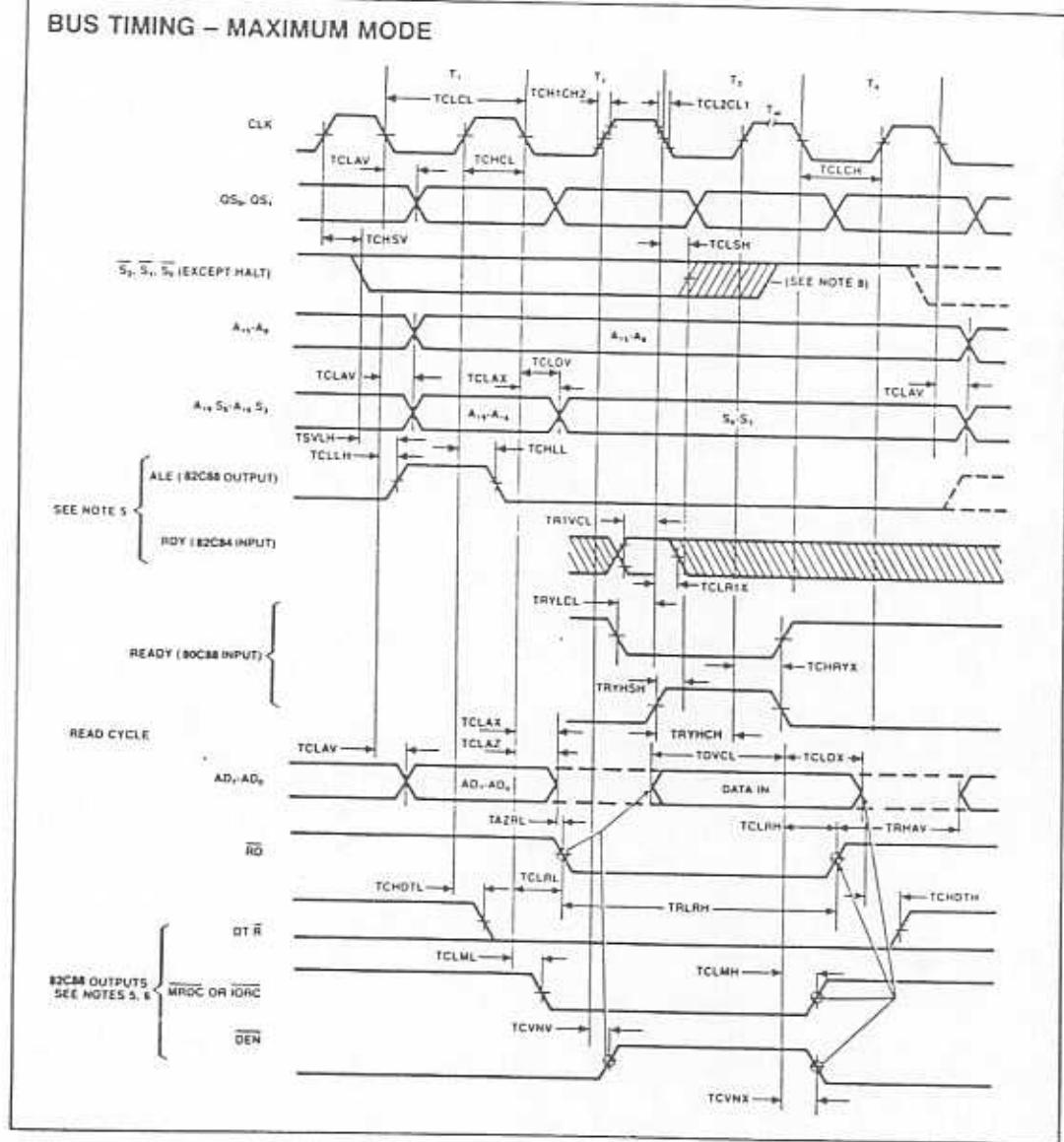
3. Applies only to T2 state (8 nanoseconds into T3).

4. The 80C88 actively pulls the RD/GT pin to a logic one on the following clock low time.

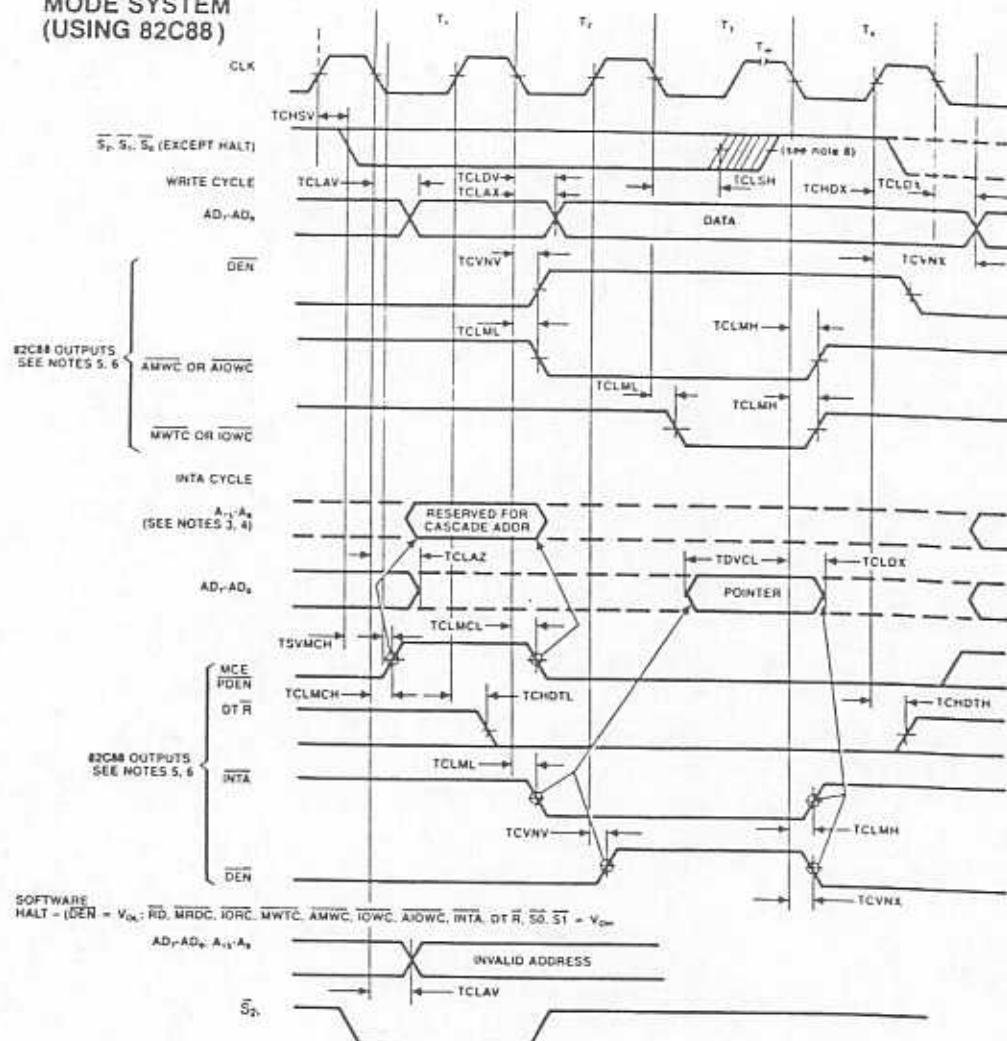
5. Status lines return to their inactive (logic one) state after CLK goes low and READY goes high.

Waveforms

BUS TIMING – MAXIMUM MODE

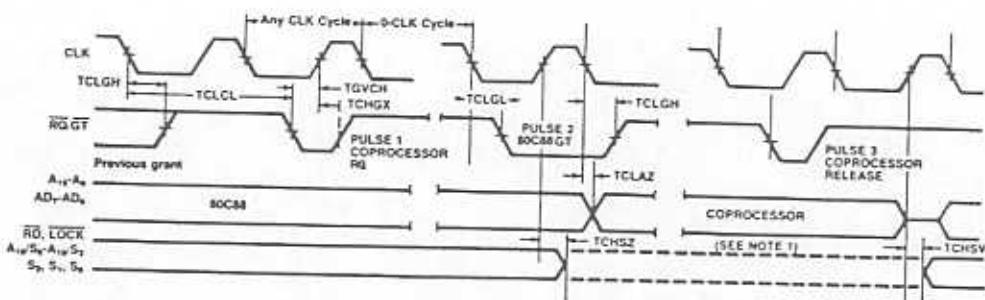


Waveforms (Continued)

BUS TIMING – MAXIMUM MODE SYSTEM
(USING 82C88)

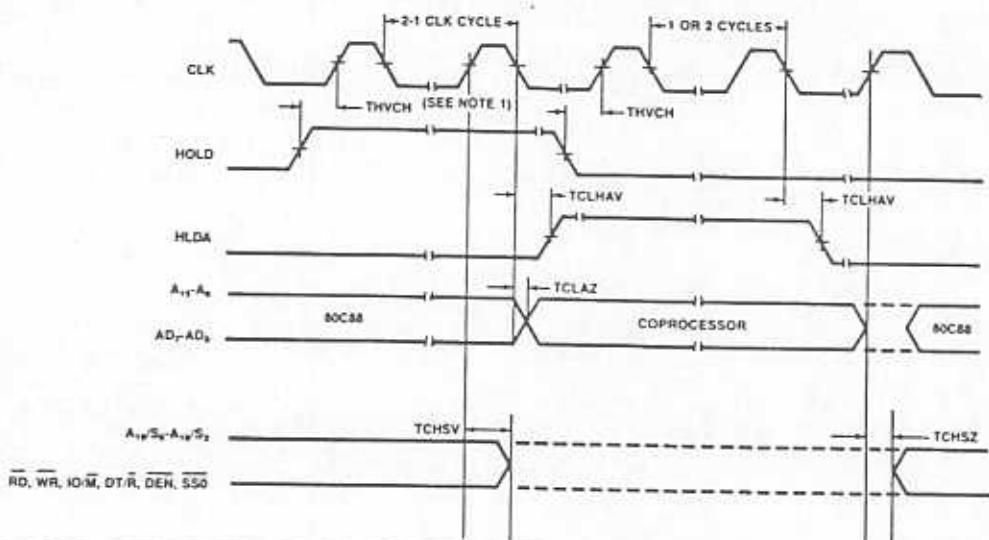
- NOTES:
- ALL SIGNALS SWITCH BETWEEN V_{OH} AND V_{OL} UNLESS OTHERWISE SPECIFIED.
 - RDY IS SAMPLED NEAR THE END OF T_2 , T_3 , T_W TO DETERMINE IF T_W MACHINES STATES ARE TO BE INSERTED.
 - CASCADE ADDRESS IS VALID BETWEEN FIRST AND SECOND INTA CYCLES.
 - TWO INTA CYCLES RUN BACK-TO-BACK. THE 80C88 LOCAL ADDR/DATA BUS IS FLOATING DURING BOTH INTA CYCLES. CONTROL FOR POINTER ADDRESS IS SHOWN FOR SECOND INTA CYCLE.
 - SIGNALS AT 82C84A OR 82C88 ARE SHOWN FOR REFERENCE ONLY.
 - THE ISSUANCE OF THE 82C88 COMMAND AND CONTROL SIGNALS (MRDC, MWTC, AMWC, IORC, IOWC, AIOWC, INTA AND DEN) LAGS THE ACTIVE HIGH 82C88 CEN.
 - ALL TIMING MEASUREMENTS ARE MADE AT 1.5V UNLESS OTHERWISE NOTED.
 - STATUS INACTIVE IN STATE JUST PRIOR TO T_4 .

REQUEST/GANT SEQUENCE TIMING (MAXIMUM MODE ONLY)

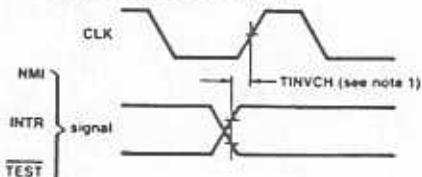


NOTE: 1. THE COPROCESSOR MAY NOT DRIVE THE BUSSES OUTSIDE THE REGION SHOWN WITHOUT RISKING CONTENTION

HOLD/HOLD ACKNOWLEDGE TIMING (MINIMUM MODE ONLY)

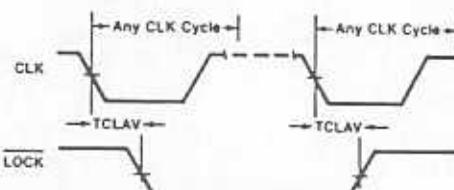


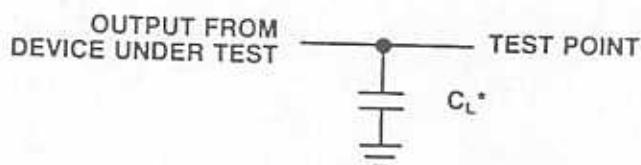
ASYNCHRONOUS SIGNAL RECOGNITION



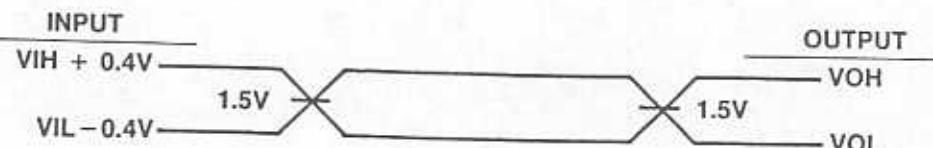
NOTE: 1. SETUP REQUIREMENTS FOR ASYNCHRONOUS SIGNALS ONLY TO GUARANTEE RECOGNITION AT NEXT CLK.

BUS LOCK SIGNAL TIMING (MAXIMUM MODE ONLY)



A.C. Test Circuits

*Includes stray and jig capacitance

A.C. Testing Input, Output Waveform

A.C. Testing: All input signals (other than CLK) must switch between $V_{IL_{max}} - 0.4V$ and $V_{IH_{min}} + 0.4V$. CLK must switch between 0.4V and $V_{CC} - 0.4V$. T_R and T_F must be less than or equal to 15ns. CLK T_R and T_F must be less than or equal to 10ns.

Instruction Set Summary

DATA TRANSFER

MOV - Move

	78542218	78542218	78542218	78542218
Register/memory to from register	1000100 w	mod reg imm		
Immediate to register/memory	1100211 w	mod 000 imm	data	data if n = 1
Immediate to register	10111 imm	data	data if n = 1	
Memory to accumulator	1010000 w	add16w	add16h	
Accumulator to Memory	1010001 w	add16w	add16h	
Register/memory to segment register	100011100	mod 0 reg imm		
Segment register to register/memory	100011000	mod 0 reg imm		

PUSH - Push

	78542218	78542218	78542218	78542218
Register/memory	11111111	mod 110 imm		
Register	01010 reg			
Segment register	000 reg 110			

POP - Pop

	78542218	78542218	78542218	78542218
Register/memory	100011111	mod 000 imm		
Register	01011 imm			
Segment register	000 reg 111			

XCHG - Exchange

	78542218	78542218	78542218	78542218
Register/memory with register	10000111 w	mod reg imm		
Register with accumulator	10010 reg			

IN - Input from

	78542218	78542218	78542218	78542218
Fixed port	1110010 w	port		
Variable port	1110110 w			

OUT - Output to

	78542218	78542218	78542218	78542218
Fixed port	1110011 w	port		
Variable port	1110111 w			
SLAT-Translate byte to AL	110101111			
LXI-Load EA to register	100011011	mod reg imm		
LXI-Load pointer to DS	110001011	mod reg imm		
LXI-Load pointer to ES	110001000	mod reg imm		
LXI-Load AH with flags	100111111			
SAHF - Store AH into flags	100111110			
PUSH - Push Flags	100111100			
POPF - Pop Flags	100111011			

ARITHMETIC

ADD - Add

	78542218	78542218	78542218	78542218
Reg/memory with register to either	0000000 w	mod reg imm		
Immediate to register/memory	1000001 w	mod 000 imm	data	data if n = 01
Immediate to accumulator	0000010 w	data	data if n = 01	

ADC - Add with carry

	78542218	78542218	78542218	78542218
Reg/memory with register to either	000010000 w	mod reg imm		
Immediate to register/memory	1000001 w	mod 010 imm	data	data if n = 01
Immediate to accumulator	000010100 w	data	data if n = 01	

INC - Increment

	78542218	78542218	78542218	78542218
Register/memory	1111111 w	mod 000 imm		
Register	01000 reg			
AAS + SCI adjust for add	001101111			
AAA + Decimal adjust for add	001001111			

SUB - Subtract

	78542218	78542218	78542218	78542218
Reg/memory and register to either	0001010 w	mod reg imm		
Immediate from register/memory	1000001 w	mod 101 imm	data	data if n = 01
Immediate from accumulator	000101100 w	data	data if n = 01	

SBB - Subtract with borrow

	78542218	78542218	78542218	78542218
Reg/memory and register to either	0001100 w	mod reg imm		
Immediate from register/memory	1000001 w	mod 011 imm	data	data if n = 01
Immediate from accumulator	0001110 w	data	data if n = 01	

Mnemonics ©Intel, 1978

DEC - Decrement

	78542218	78542218	78542218	78542218
Register/memory	1111111 w	mod 011 imm		
Register	01001 reg			
NEA Change sign	1111011 w	mod 011 imm		

CMP - Compare

	78542218	78542218	78542218	78542218
Register/memory and register	0011100 w	mod reg imm		
Immediate with register/memory	1000021 w	mod 111 imm	data	data if n = 01
Immediate with accumulator	0011110 w	data	data if n = 01	
AAA ASCII adjust for subtract	0011111			
AAA Decimal adjust for subtract	0110111			
MUL - Multiply unsigned	1111011 w	mod 120 imm		
MULH - Integer multiply unsigned	1111010 w	mod 011 imm		
MULW - ASCII adjust for multiply	11010100	0000101010		
DIV - Divide unsigned	1111011 w	mod 110 imm		
IDIV - Integer divide unsigned	1111010 w	mod 111 imm		
ADIV - ASCII adjust for divide	11010101	000001010		
CRW - Convert word to word	100111000			
CWB - Convert word to double word	100111001			

LOGIC

NOT - Invert

	78542218	78542218	78542218	78542218
SHL/SAL Shift logical arithmetic left	1101000 w	mod 100 imm		
SHR Shift logical right	1101000 w	mod 011 imm		
SAR Shift arithmetic right	1101000 w	mod 111 imm		
ROL Rotate left	1101000 w	mod 000 imm		
RRN Rotate right	1101000 w	mod 011 imm		
RCL Rotate through carry left	1101000 w	mod 010 imm		
RCR Rotate through carry right	1101000 w	mod 011 imm		

AND - And

	78542218	78542218	78542218	78542218
Reg/memory and register to either	0010000 w	mod reg imm		
Immediate to register/memory	1000000 w	mod 010 imm	data	data if n = 01
Immediate to accumulator	0000000 w	data	data if n = 01	

OR - Or

	78542218	78542218	78542218	78542218
Reg/memory and register to either	0000100 w	mod reg imm		
Immediate to register/memory	1000000 w	mod 001 imm	data	data if n = 01
Immediate to accumulator	0000000 w	data	data if n = 01	

STRING MANIPULATION

	78542218	78542218	78542218	78542218
REP - Repeat	111100111			
MOV - Move byte/word	101000100			
CMPS - Compare byte/word	101001111			
SCAS - Scan byte/word	101011111			
LODS - Load string to AL/AH	101011000			
STOS - Store byte/word from AL/AH	101010101			

Instruction Set Summary (continued)

CONTROL TRANSFER

CALL - Call

Direct within segment
Indirect within segment

78543210 78543210 78543210		
11101000	disp-low	disp-high
11111111	mod 010 r/m	
10011010	offset-low	offset-high
	seg-low	seg-high
11111111	mod 011 r/m	

JMP - Unconditional Jump

Direct within segment
Direct within segment short
Indirect within segment
Direct intersegment

78543210 78543210 78543210		
11101001	disp-low	disp-high
11101011	disp	
11111111	mod 100 r/m	
11101010	offset-low	offset-high*
	seg-low	seg-high
11111111	mod 101 r/m	

RET - Return from CALL

Within segment

78543210 78543210 78543210		
11000011		
11000010	data-low	data-high
11001011		
11001010	data-low	data-high
01110100	disp	
01111100	disp	
01111110	disp	
01110010	disp	
01110110	disp	
01111010	disp	
01110000	disp	
01111000	disp	
01110101	disp	
01111101	disp	
01111111	disp	

Intersegment

Intersegment adding immediate to SP:

JE/JZ - Jump on equal/zero

JL/JGE - Jump on less/not greater or equal

JL/JGE - Jump on less or equal/not greater

JBE/JNAE - Jump on below/not above

JBE/JNAE - Jump on below or equal/not above

JP/JPE - Jump on parity/parity even

JB - Jump on overflow

JS - Jump on sign

JNE/JNZ - Jump on not equal/not zero

JNL/JGE - Jump on not less/greater or equal

JBLE/JSG - Jump on not less or equal/greater

Footnotes:

AL = 8-bit accumulator

AX = 16-bit accumulator

CX = Count register

DS = Data segment

ES = Extra segment

Above/below refers to unsigned value

Greater = more positive

Less = less positive (more negative) signed values

if d = 1 then "to" reg; if d = 0 then "from" reg

if w = 1 then word instruction; if w = 0 then byte instruction

if mod = 11 then r/m is treated as a REG field

if mod = 00 then DISP = 0*, disp-low and disp-high are absent

if mod = 01 then DISP = disp-low sign-extended to 16-bits; disp-high is absent

if mod = 10 then DISP = disp-high disp-low

if r/m = 000 then EA = (BX) + (SI) + DISP

if r/m = 001 then EA = (BX) + (DI) + DISP

if r/m = 010 then EA = (BP) + (SI) + DISP

if r/m = 011 then EA = (BP) + (DI) + DISP

if r/m = 100 then EA = (SI) + DISP

if r/m = 101 then EA = (DI) + DISP

if r/m = 110 then EA = (BP) + DISP*

if r/m = 111 then EA = (BX) + DISP

DISP follows 2nd byte of instruction (before data if required)

*except if mod = 00 and r/m = 110 then EA = disp-high/ disp-low

JBB/JAE - Jump on not below/above or equal

JBE/JA - Jump on not below or equal/above

JBP/JPS - Jump on not pair/par odd

JBO - Jump on not overflow

JBS - Jump on not sign

LOOP/LOOPR - Loop while zero/equal

LOOPN/LOOPNR - Loop while not zero/less

JCZ/JZ - Jump on Cx zero

78543210 78543210

01110011 010

01110111 disp

01111011 disp

01110001 disp

11100001 disp

11100000 disp

11100011 disp

INT - Interrupt

Type specified

Type 3

INTO - Interrupt on overflow

IRET - Interrupt return

11001101 type

11001100

11001110

11001111

PROCESSOR CONTROL

CLC - Clear carry

CMC - Complement carry

STC - Set carry

CLD - Clear direction

STD - Set direction

CLI - Clear interrupt

STI - Set interrupt

HLT - Halt

WAI - Wait

ESC - Escape to external device

LOCK - Bus lock prefix

11111000

11110101

11111001

11111100

11111101

11111010

11111011

11111000

11011011 mod x z s r/m

11111000

if s w = 01 then 16 bits of immediate data form the operand

if s w = 11 then an immediate data byte is sign extended to form the 16-bit operand

if v = 0 then "count" = 1, if v = 1 then "count" in CL

x = don't care

z is used for string primitives for comparison with ZF FLAG

SEGMENT OVERRIDE PREFIX

00 0 1 reg 1 1 0

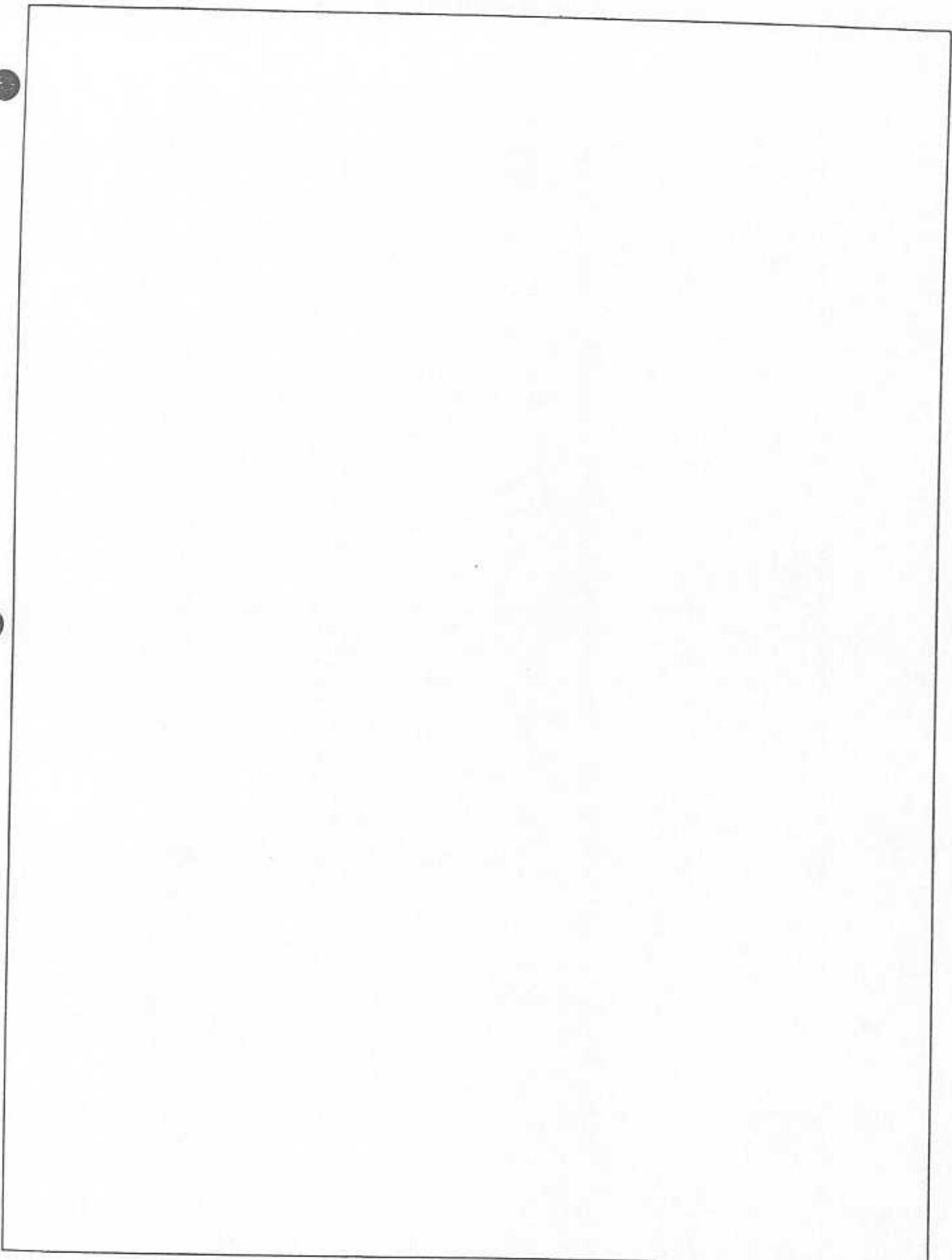
REG is assigned according to the following table

16-Bit [w = 1]	8-Bit [w = 0]	Segment
000 AX	000 AL	00 ES
001 CX	001 CL	01 CS
010 DX	010 DL	10 SS
011 BX	011 BL	11 DS
100 SP	100 AH	
101 BP	101 CH	
110 SI	110 DH	
111 DI	111 BH	

Instructions which reference the flag register file as a 16-bit object use the symbol FLAGS to represent the file.

FLAGS = X X X X (OF) (OF) (TF) (SF) (ZF) X (AF) X (PF) X (CF)

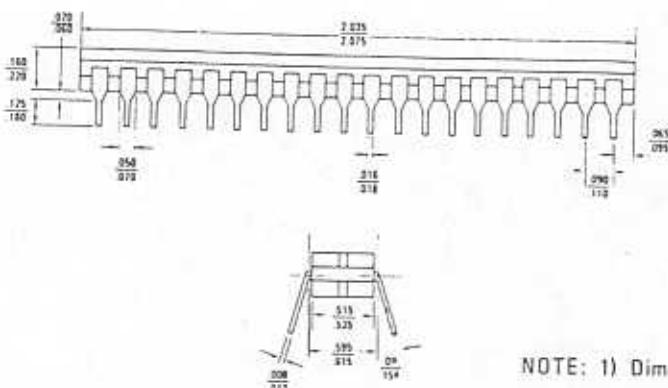
Notes



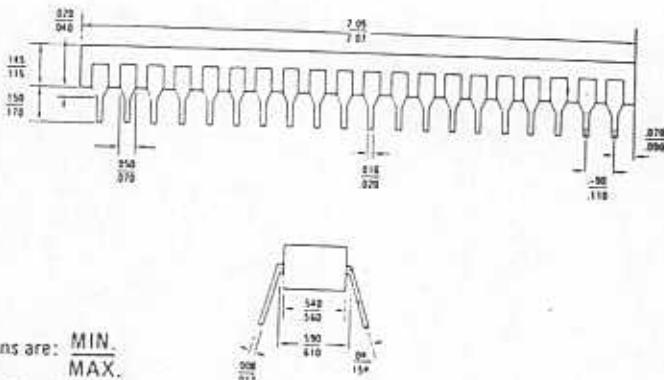
Notes

Packaging

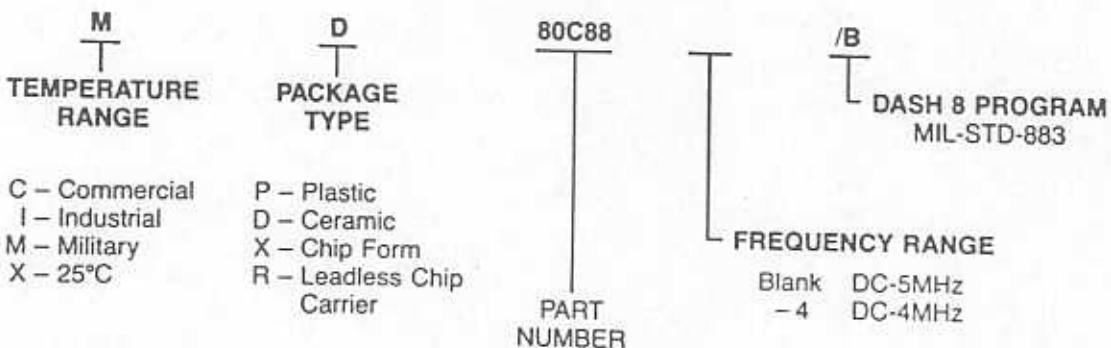
40 LEAD CERDIP



40 LEAD PLASTIC



NOTE: 1) Dimensions are:
MIN.
MAX.
2) All Dimensions in inches

Ordering Information*Sales Offices*

P.O. BOX 7065 HUNTSVILLE, AL. 35807 (205) 837-8886	SUITE 205 6400 CANOGA AVE. WOODLAND HILLS, CA. 91367 (818) 992-0686	SUITE 101 1717 EAST 116TH STREET CARMEL, IN. 46032 (317) 844-8011	SUITE 426 555 BROAD HOLLOW ROAD MELVILLE, NY. 11747 (516) 249-4500	33919 9TH AVE., SOUTH FEDERAL WAY, WA. 98003 (206) 838-4878
SUITE 250 1717 E. MORTEN AVE. PHOENIX, AZ. 85020 (602) 870-0080	SUITE 215 3890 W. COMMERCIAL BLVD. FT. LAUDERDALE, FL. 33309 (305) 739-0016	SUITE 308 1 BURLINGTON WOODS DR. BURLINGTON, MA. 01803 (617) 273-5942	SUITE 280 300 E. WILSON BRIDGE RD. WORTHINGTON, OH. 43085 (614) 885-5957	SUITE 8 2005 BROADWAY VANCOUVER, WA. 98663 (206) 696-0043
SUITE 320 1503 S.O. COAST DRIVE COSTA MESA, CA. 92626 (714) 540-2176	SUITE 400 875 JOHNSON FERRY RD. ATLANTA, GA. 30342 (404) 256-4000	SUITE 703 2850 METRO DRIVE MINNEAPOLIS, MN. 55420 (612) 854-3558	SUITE 1101 996 OLD EAGLE SCHOOL RD. WAYNE, PA. 19087 (215) 687-6680	
SUITE C100 883 STIERLIN ROAD MOUNTAIN VIEW, CA. 94043 (415) 964-6443	SUITE 300 6400 SHAFER COURT ROSEMONT, IL. 60018 (312) 692-4960	P.O. BOX 31747 RALEIGH, NC. 27622 (919) 847-8985	SUITE 110 17120 DALLAS PARKWAY DALLAS, TX. 75248 (214) 248-3237	 HARRIS

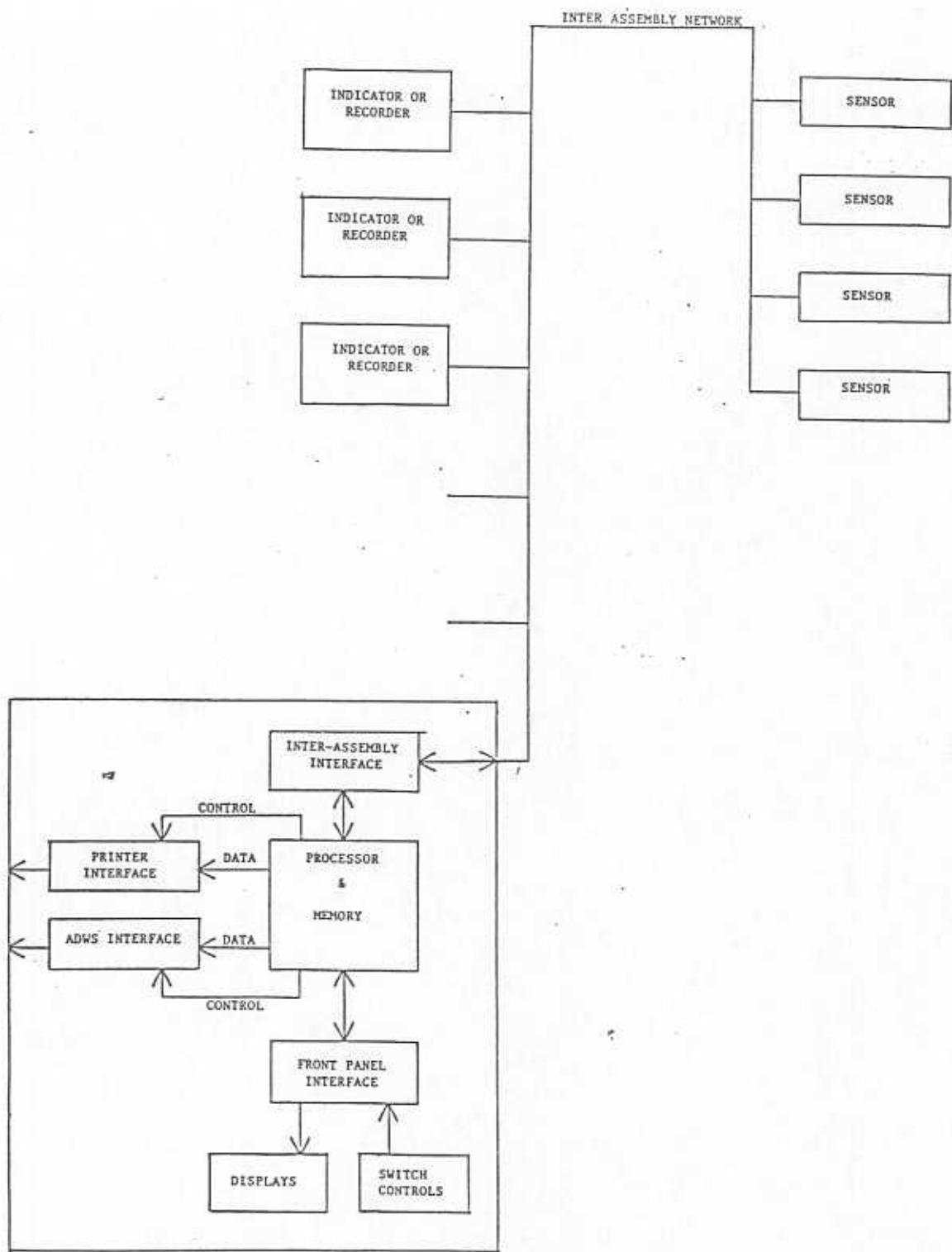
SEMICONDUCTOR DIGITAL PRODUCTS DIVISION

McClellan AFB
Contract #: F04606-85-C0733
CDRL Item: E00D
15 May 90

CHARTS AND FIGURES
FOR THE
AN/FMQ-13(V)
DIGITAL WIND SENSOR
APPLICATION PROGRAM
(DWG # 8200-1013)

Figure 1

SYSTEM BLOCK DIAGRAM



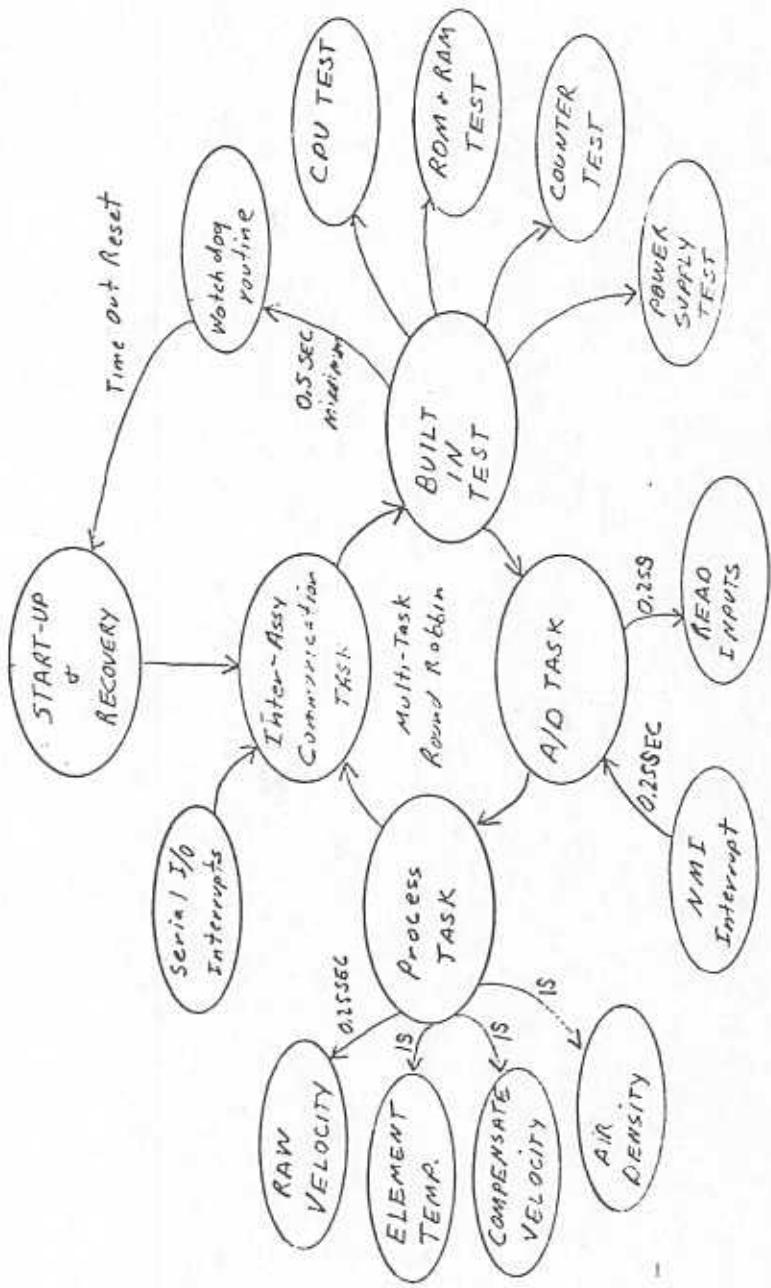


Figure 2 - Control Flow Diagram

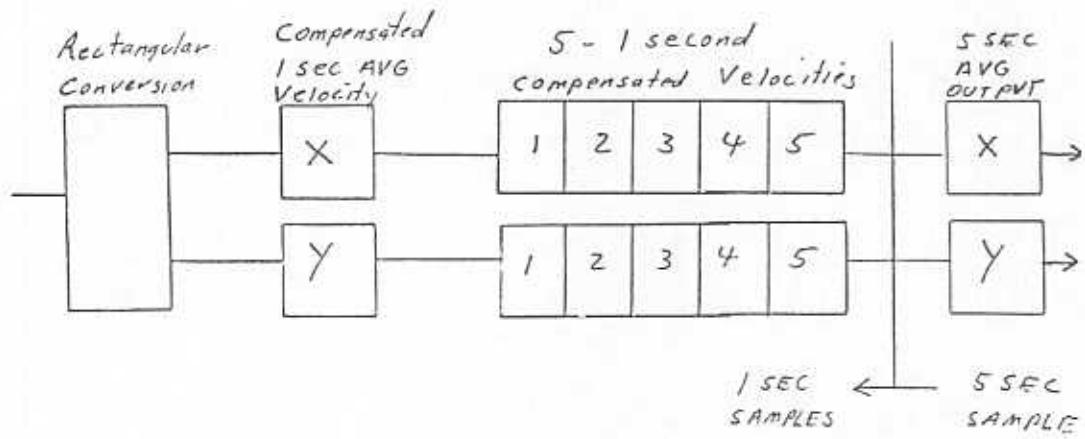
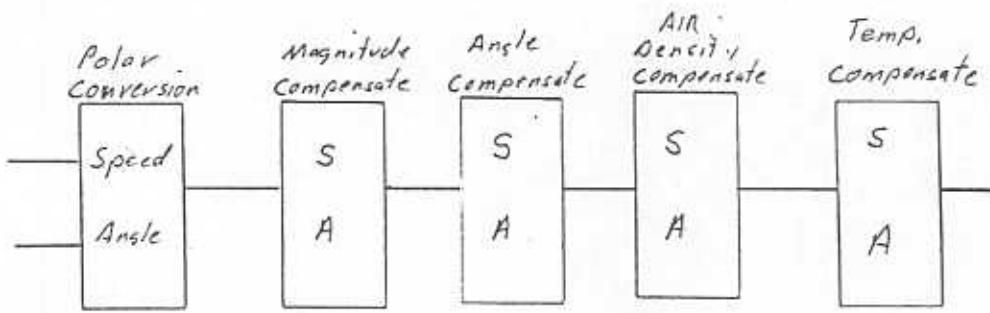
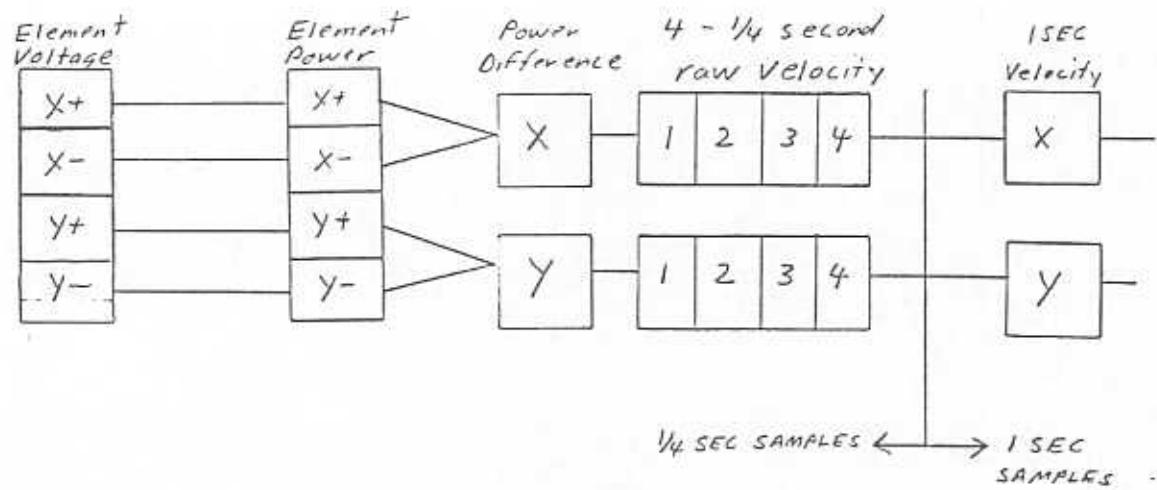


Figure 3 - Data Flow Diagram

TIMING RELATIONSHIPS

↑
250 m sec.
↓

- Even sec. DETERMINE RAW VELOCITIES
DETERMINE COMPENSATED VELOCITIES
DETERMINE AIR DENSITY
- DETERMINE RAW VELOCITIES
- DETERMINE RAW VELOCITIES
- DETERMINE RAW VELOCITIES
SET ELEMENT TEMPERATURE
- Odd sec. DETERMINE RAW VELOCITIES
DETERMINE COMPENSATED VELOCITIES
DETERMINE AIR DENSITY

Figure 4

MEMORY MAP

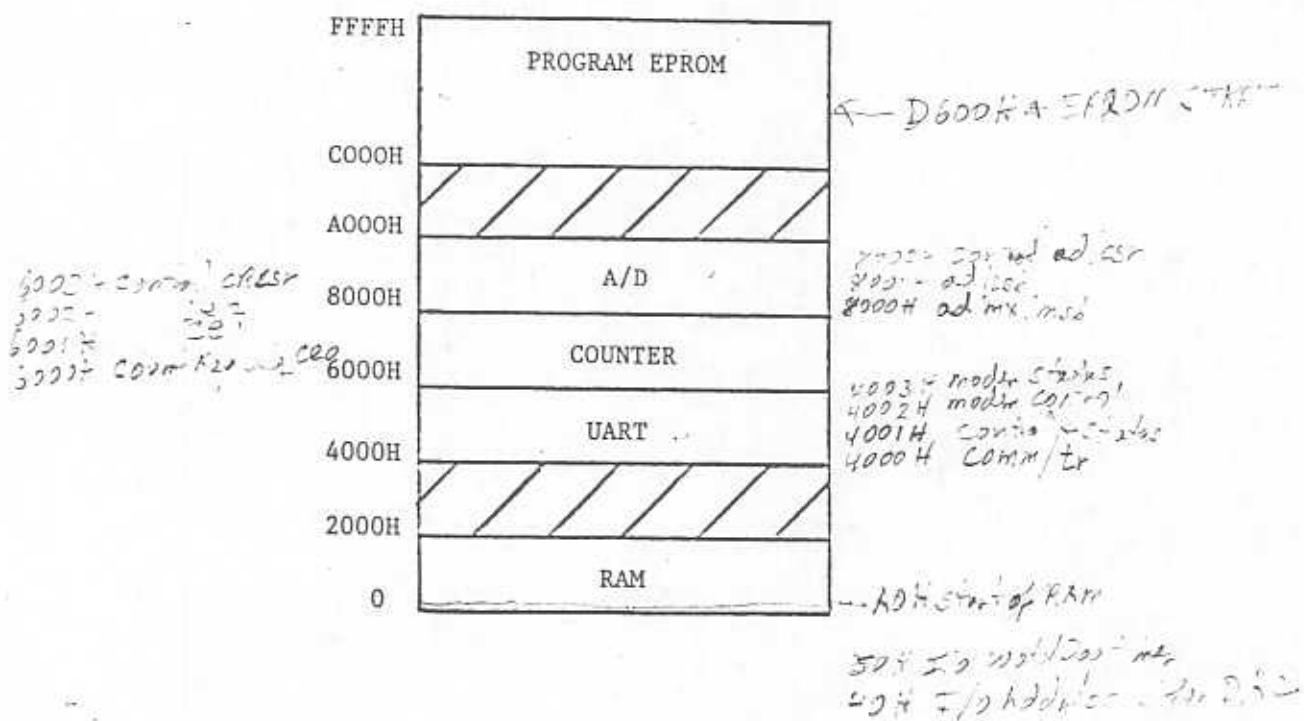


Figure 5

I/O MEMORY MAP

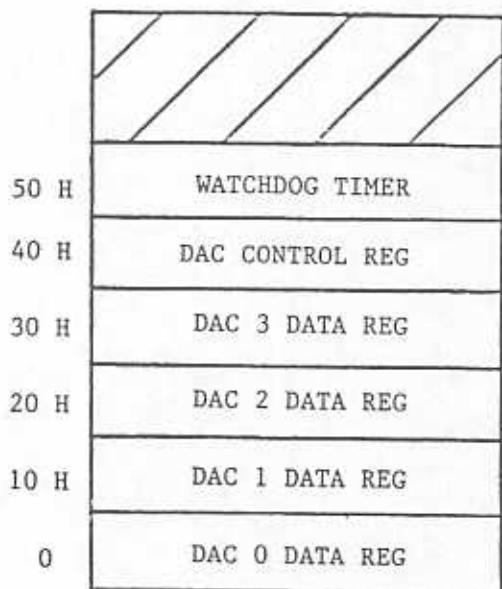


Figure 6

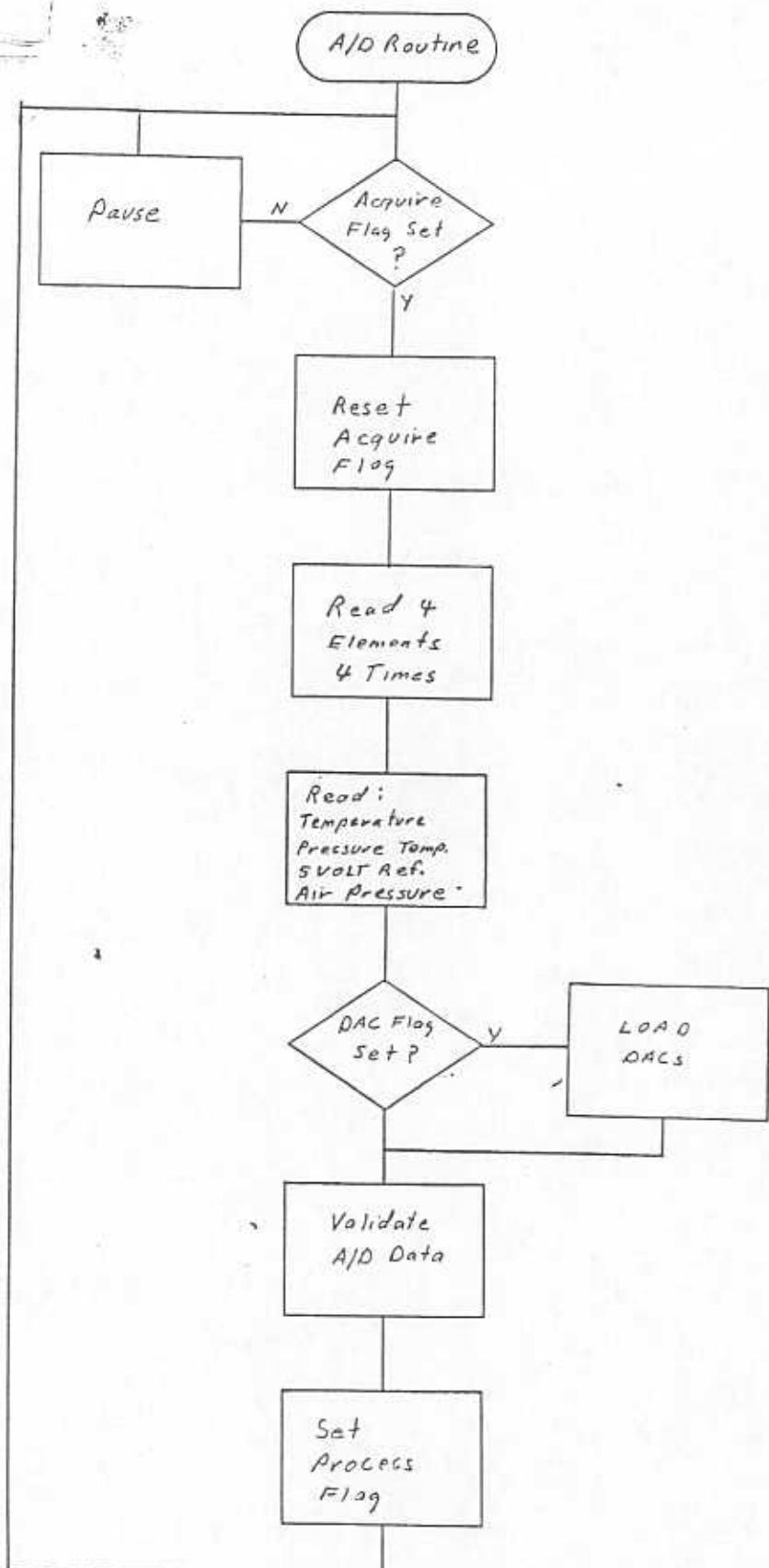


Figure 7

DETERMINE RAW VELOCITIES

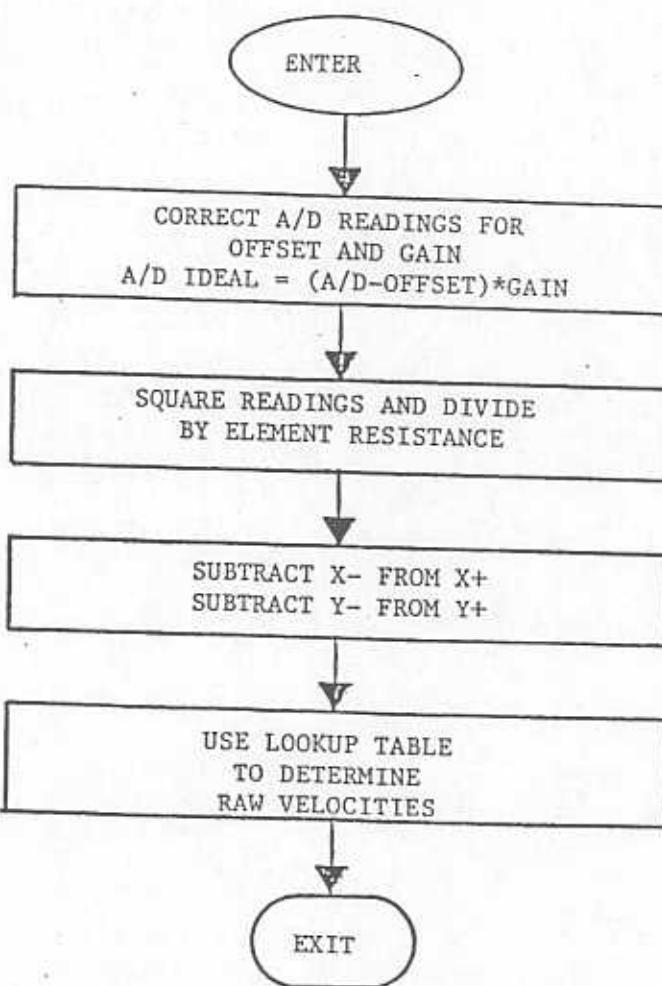


Figure 8

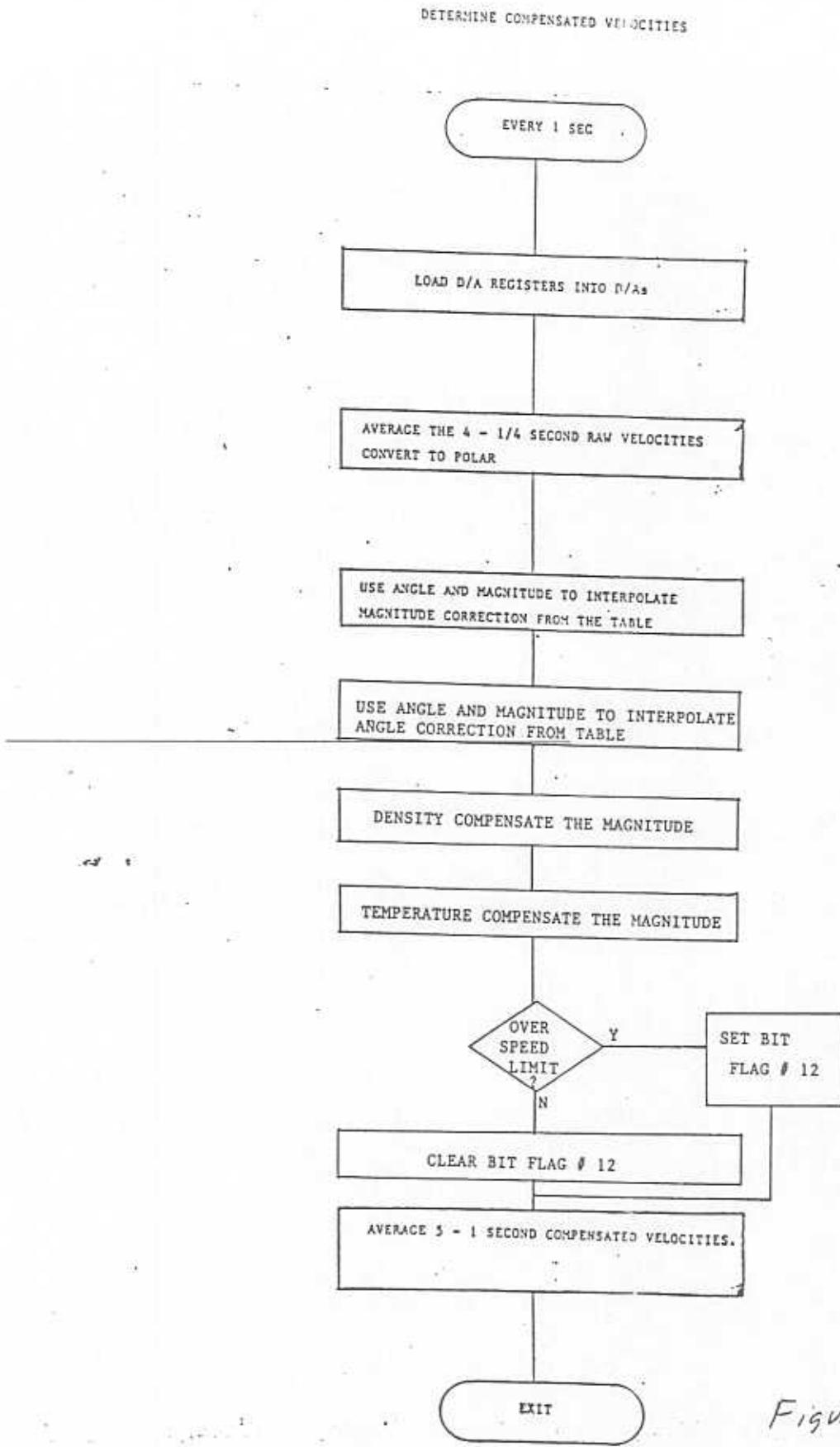


Figure - 9

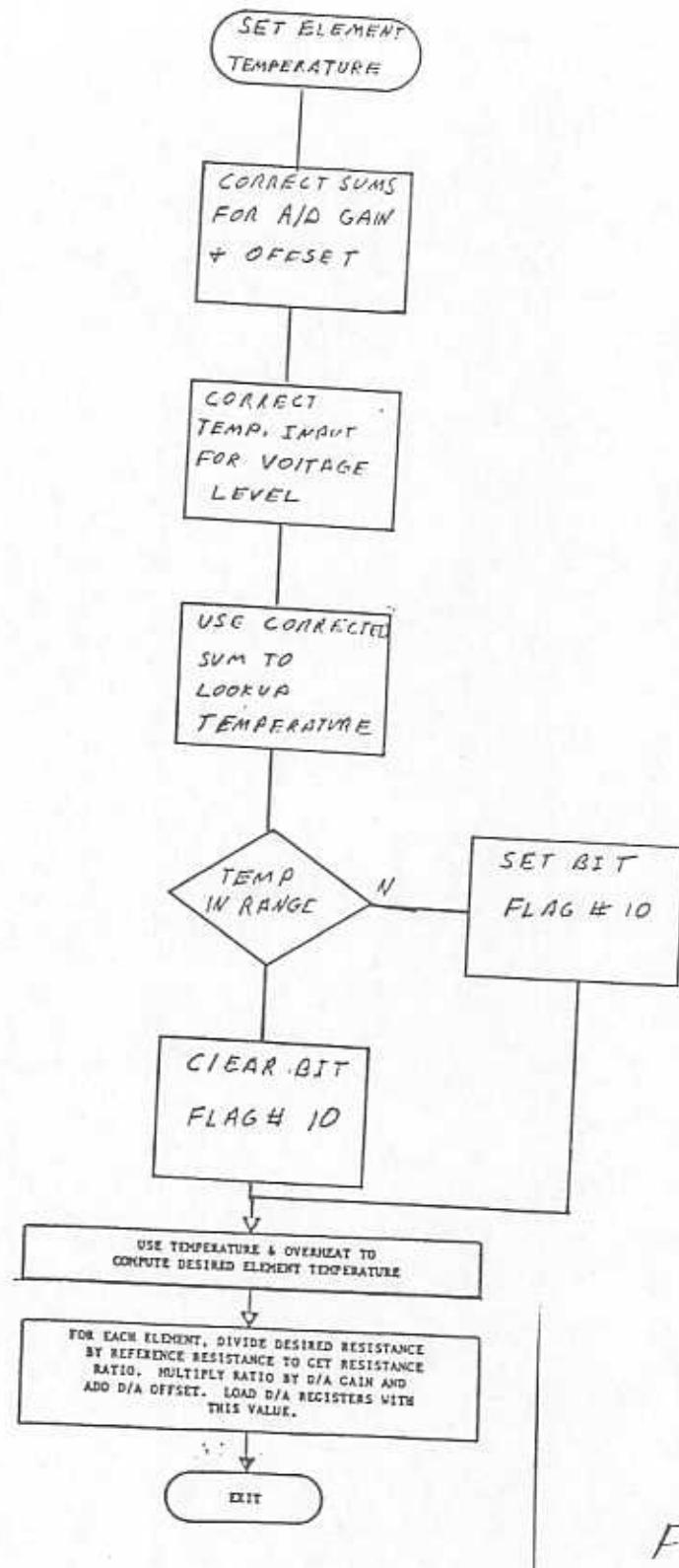


Figure -10

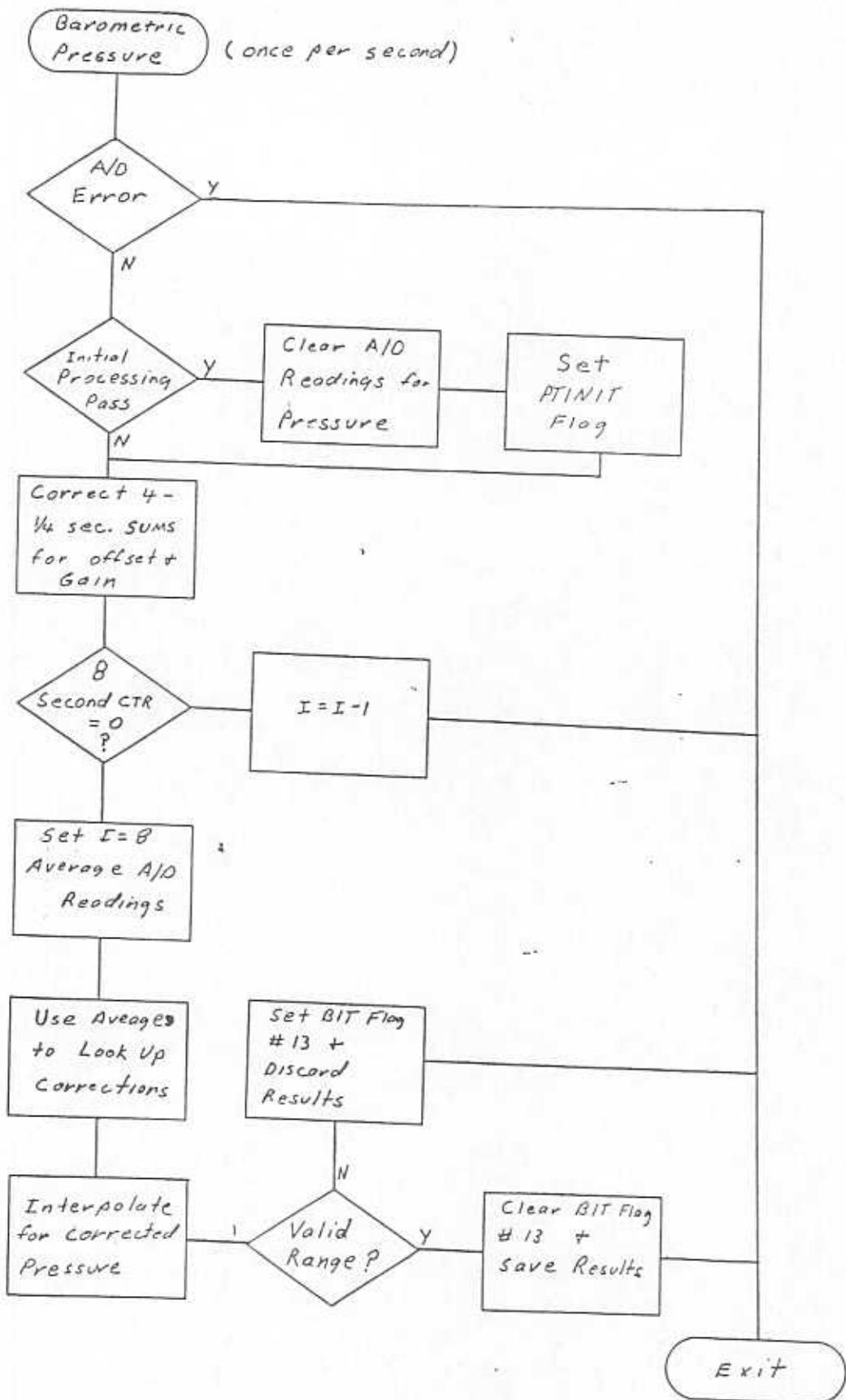


Figure 11

SENSOR INTER-ASSY COMMUNICATION TASK

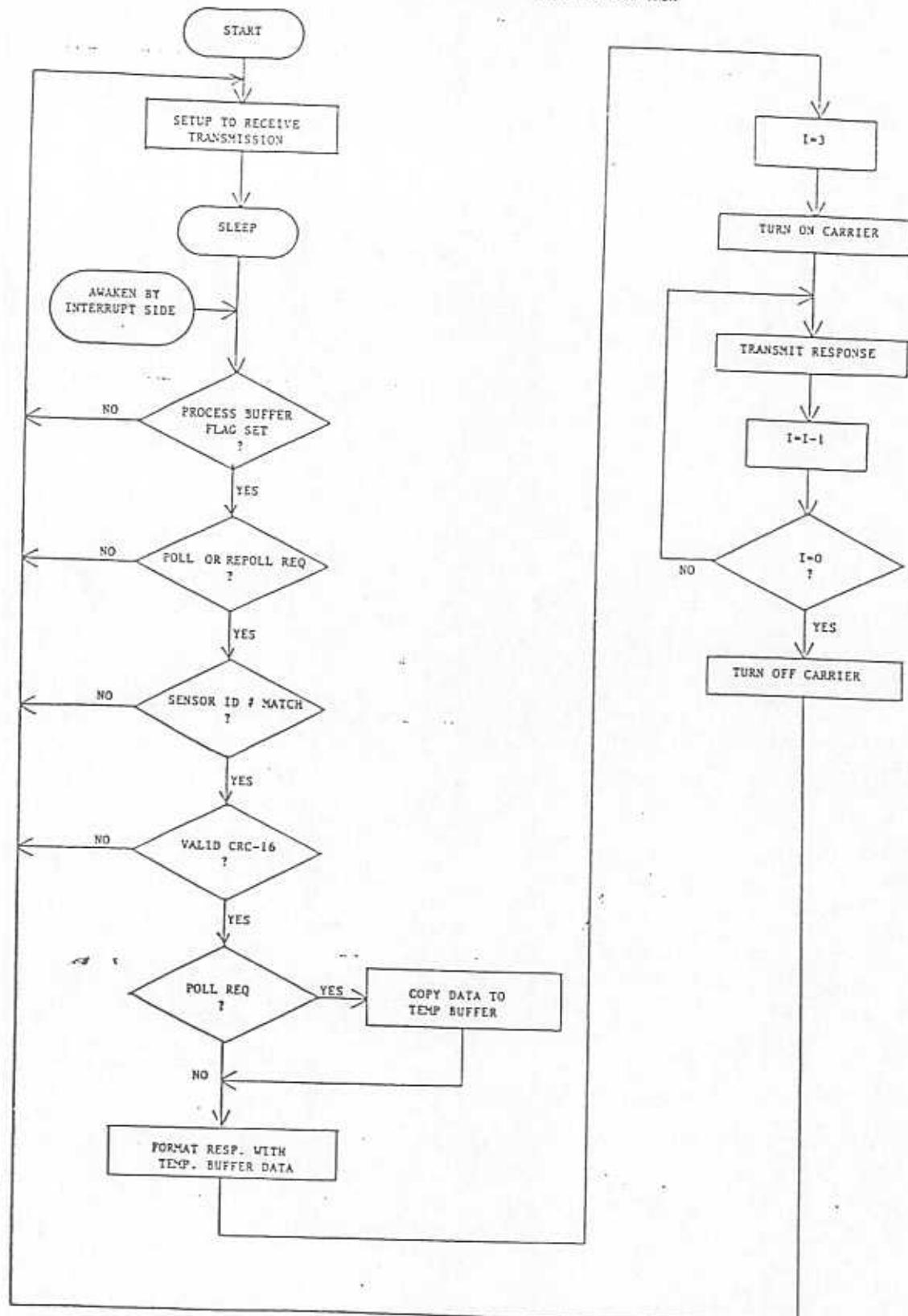


Figure 12

SENSOR INTER-ASSEMBLY COMMUNICATION INTERRUPT HANDLER

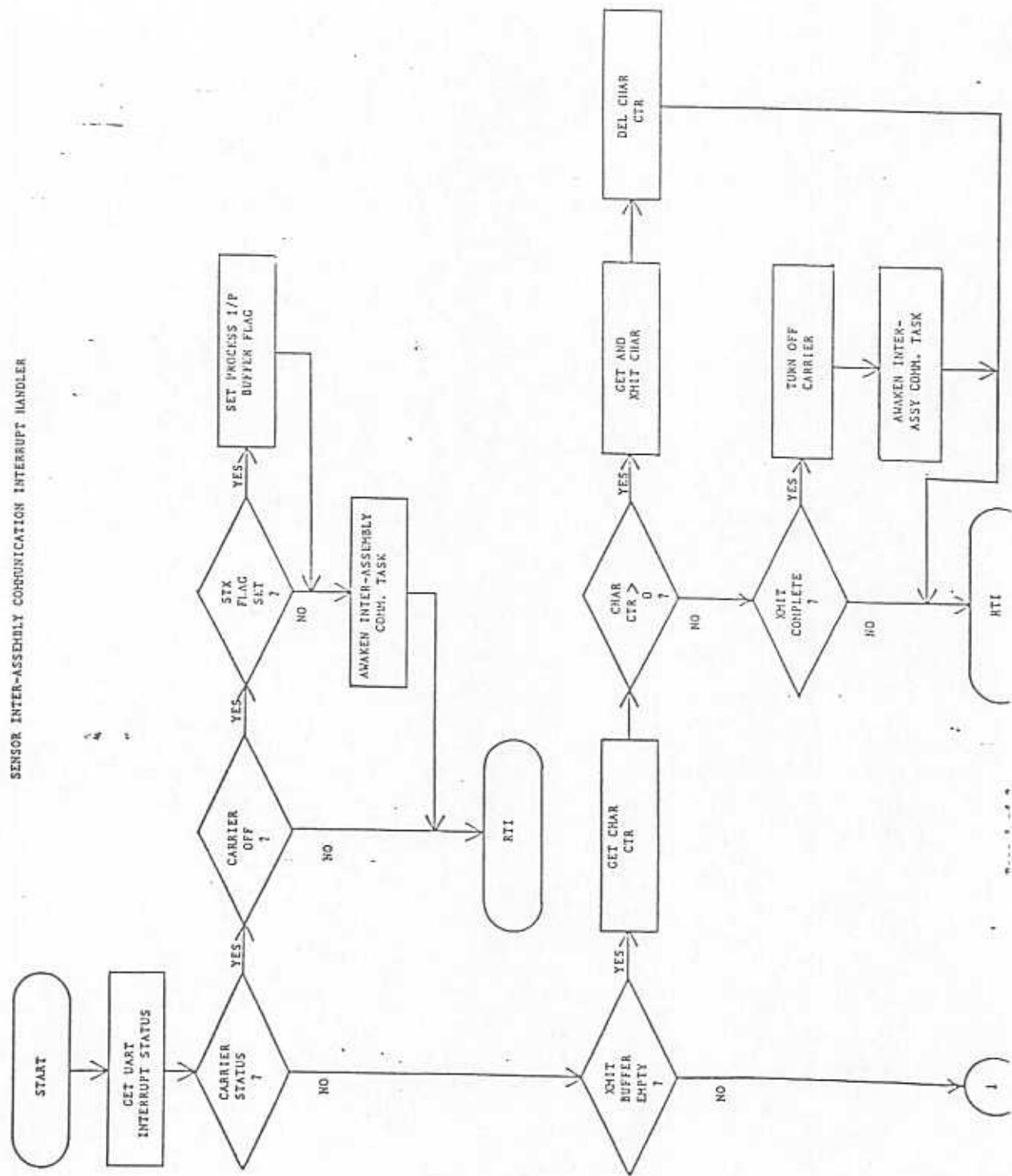


Figure 13

SENSOR INTER-ASSY COMMUNICATION INTERRUPT HANDLER

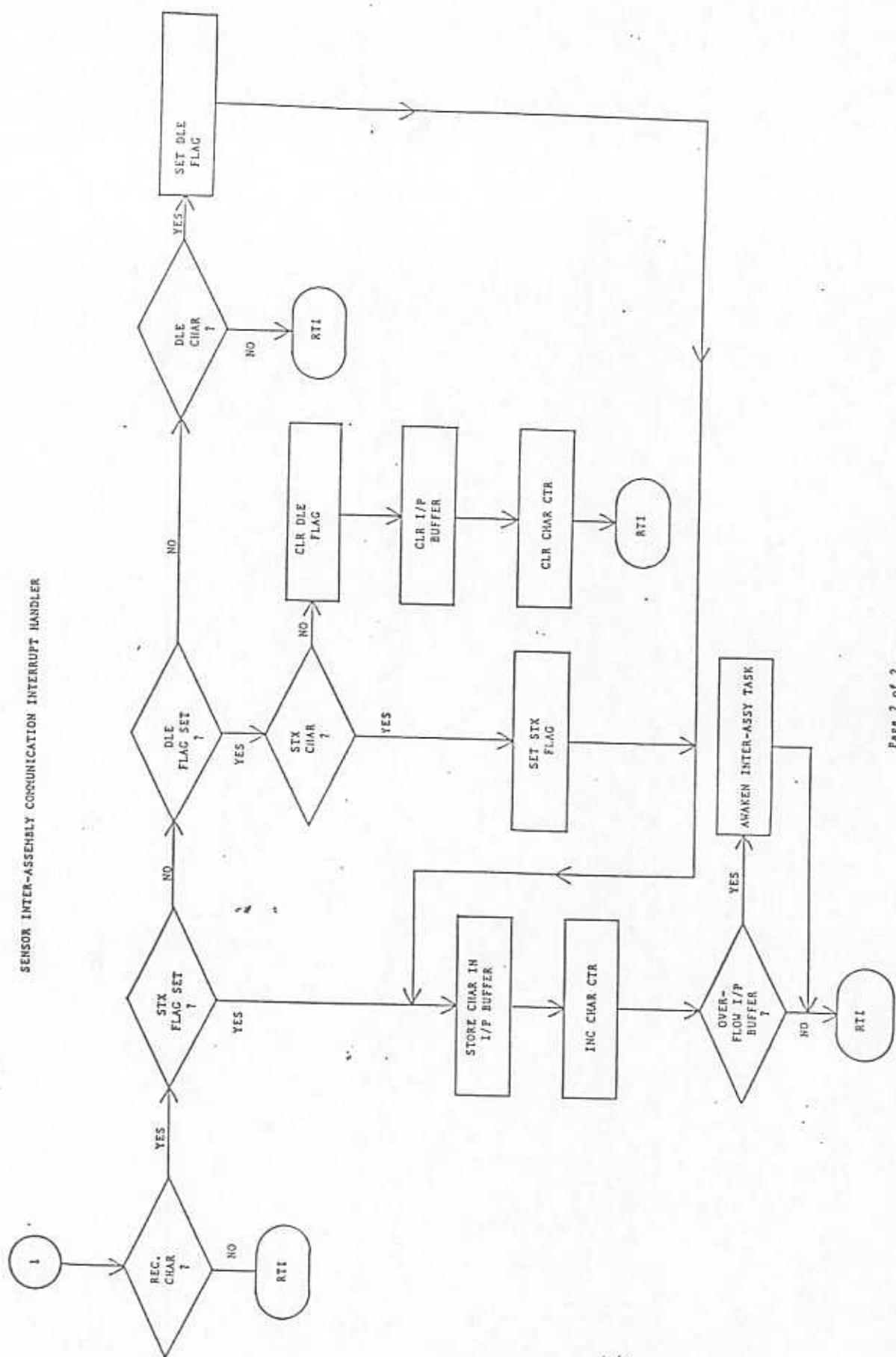


Figure 14

SENSOR STATUS WORD

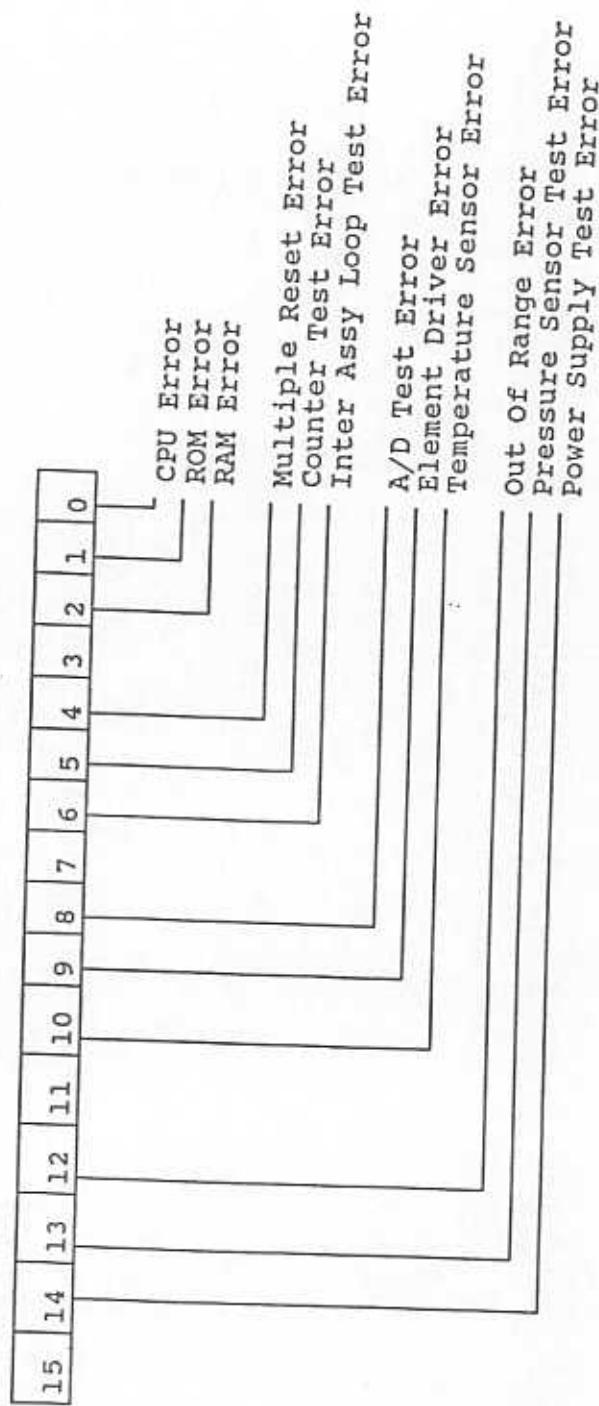


FIGURE 43 (B)

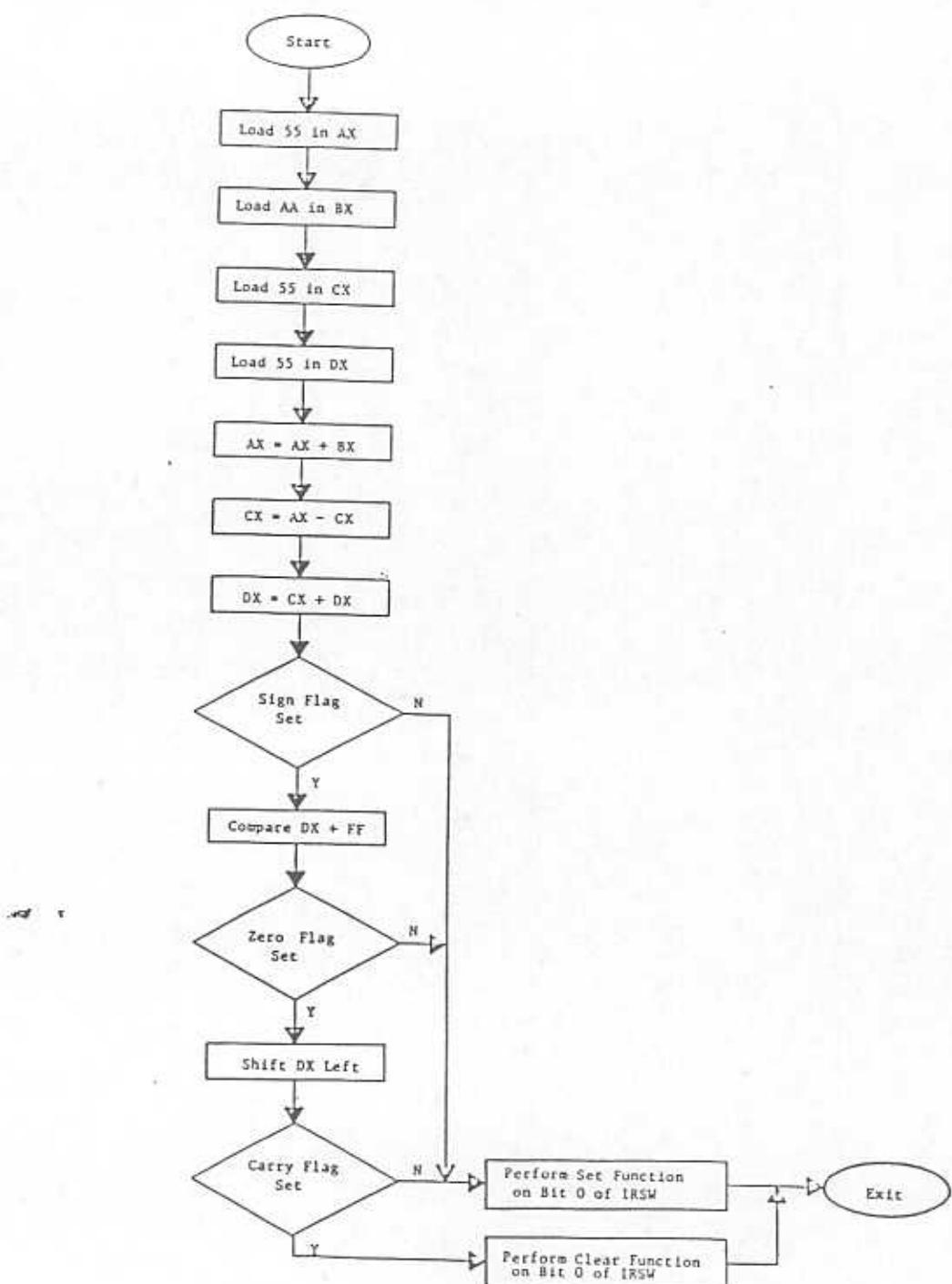
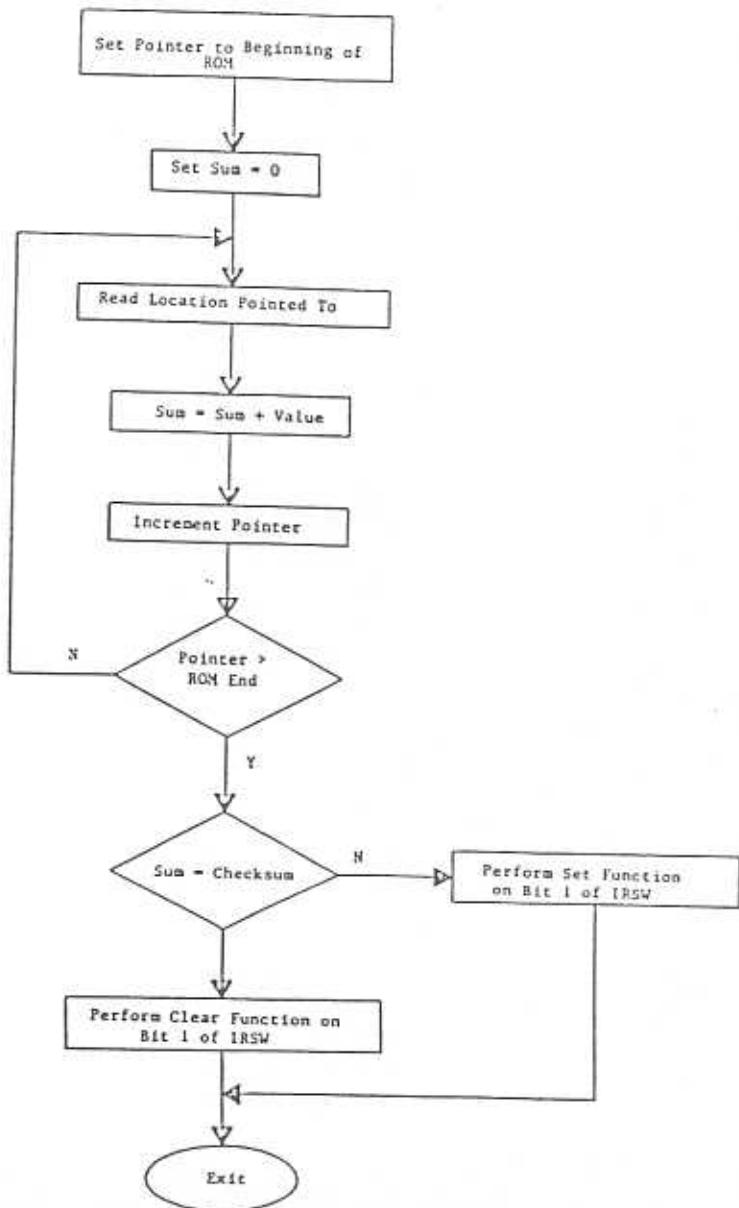


FIGURE 45

CPU TEST FLOW GRAPH

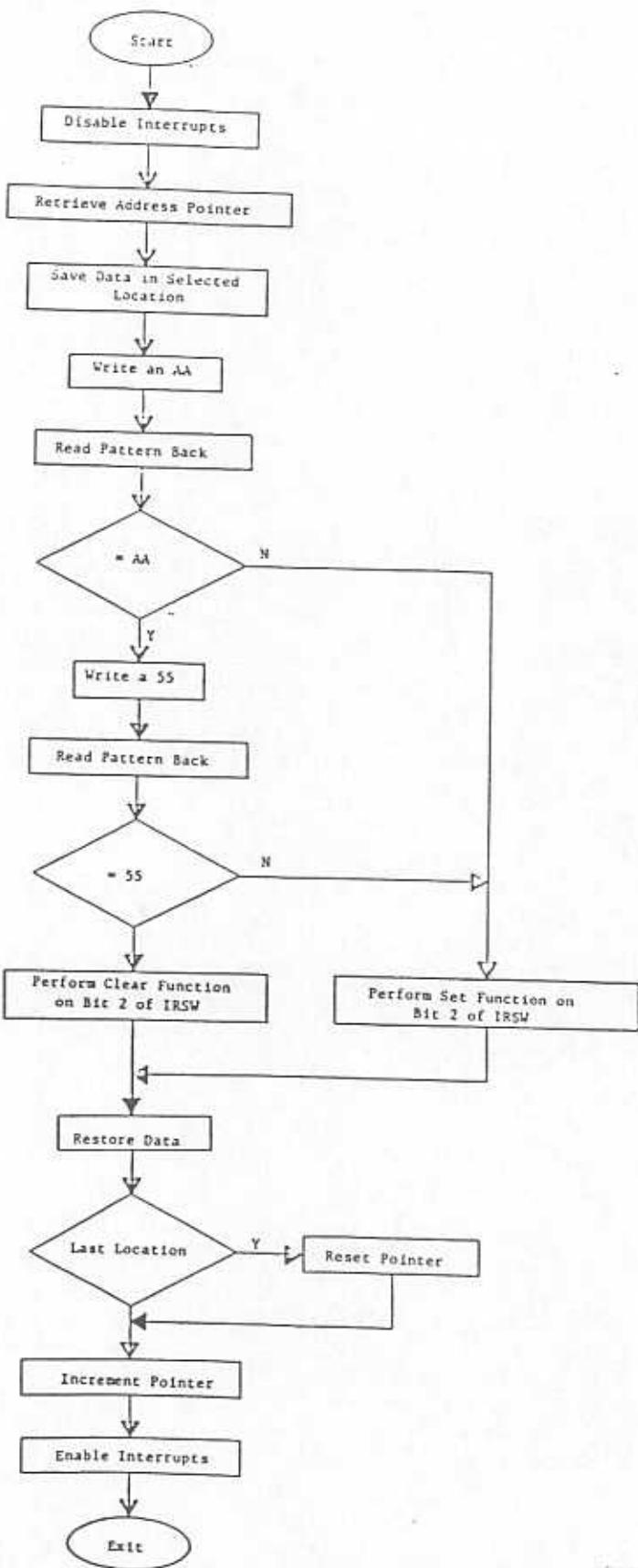


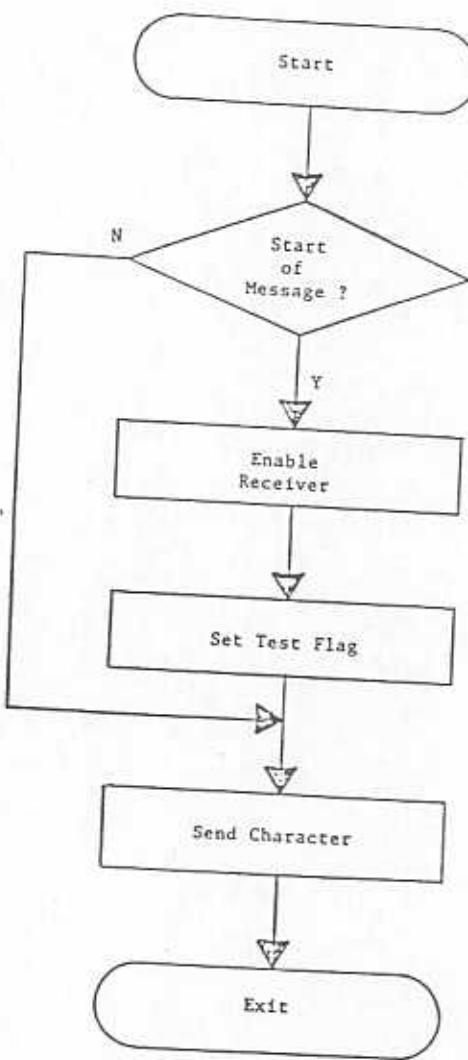
17
FIGURE 46

ROM TEST FLOW GRAPH

FIGURE 47

RAM TEST FLOW GRAPH





19
Figure 48A

IAC LOOP TEST TRANSMIT ROUTINE

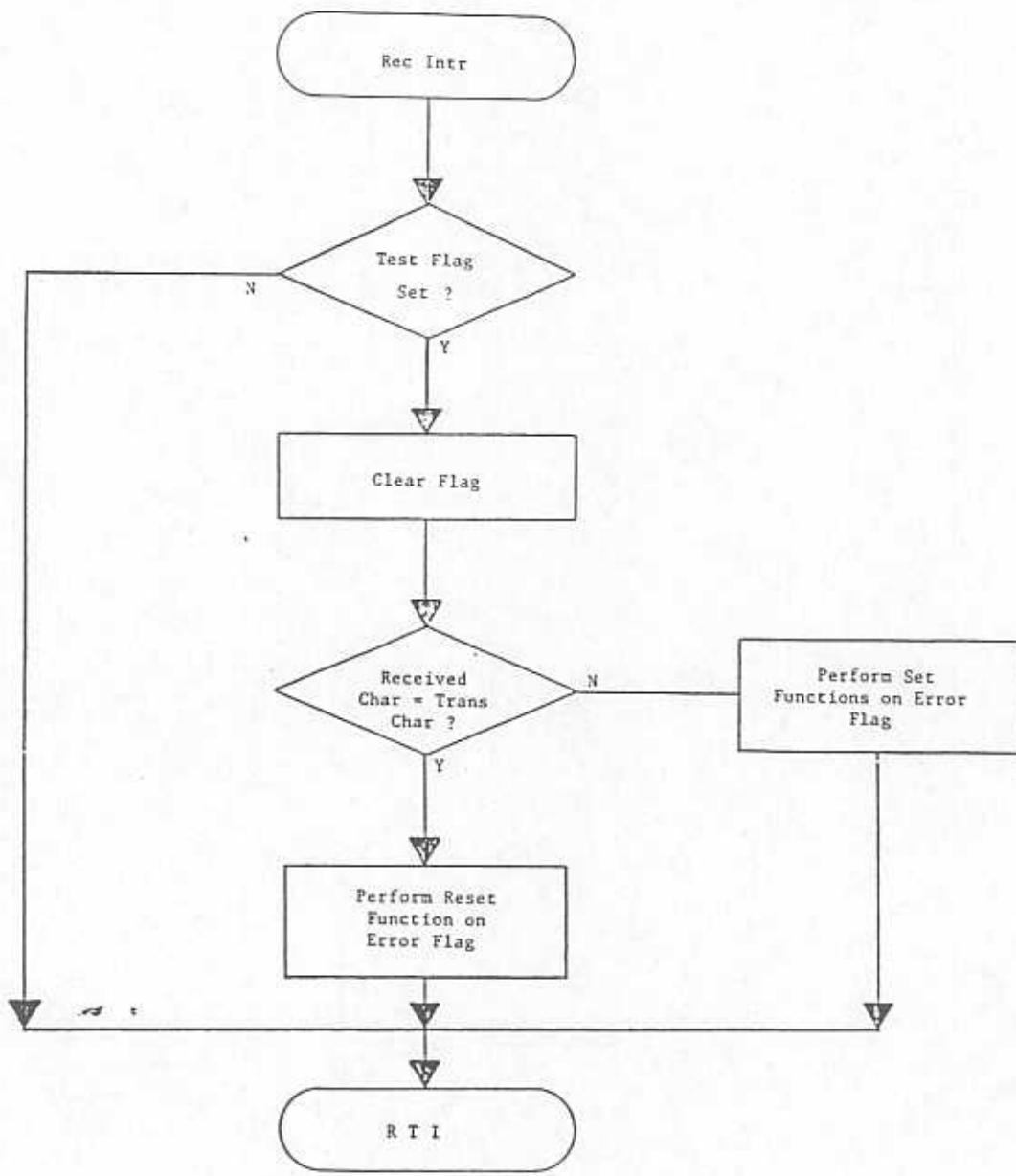


Figure 48B

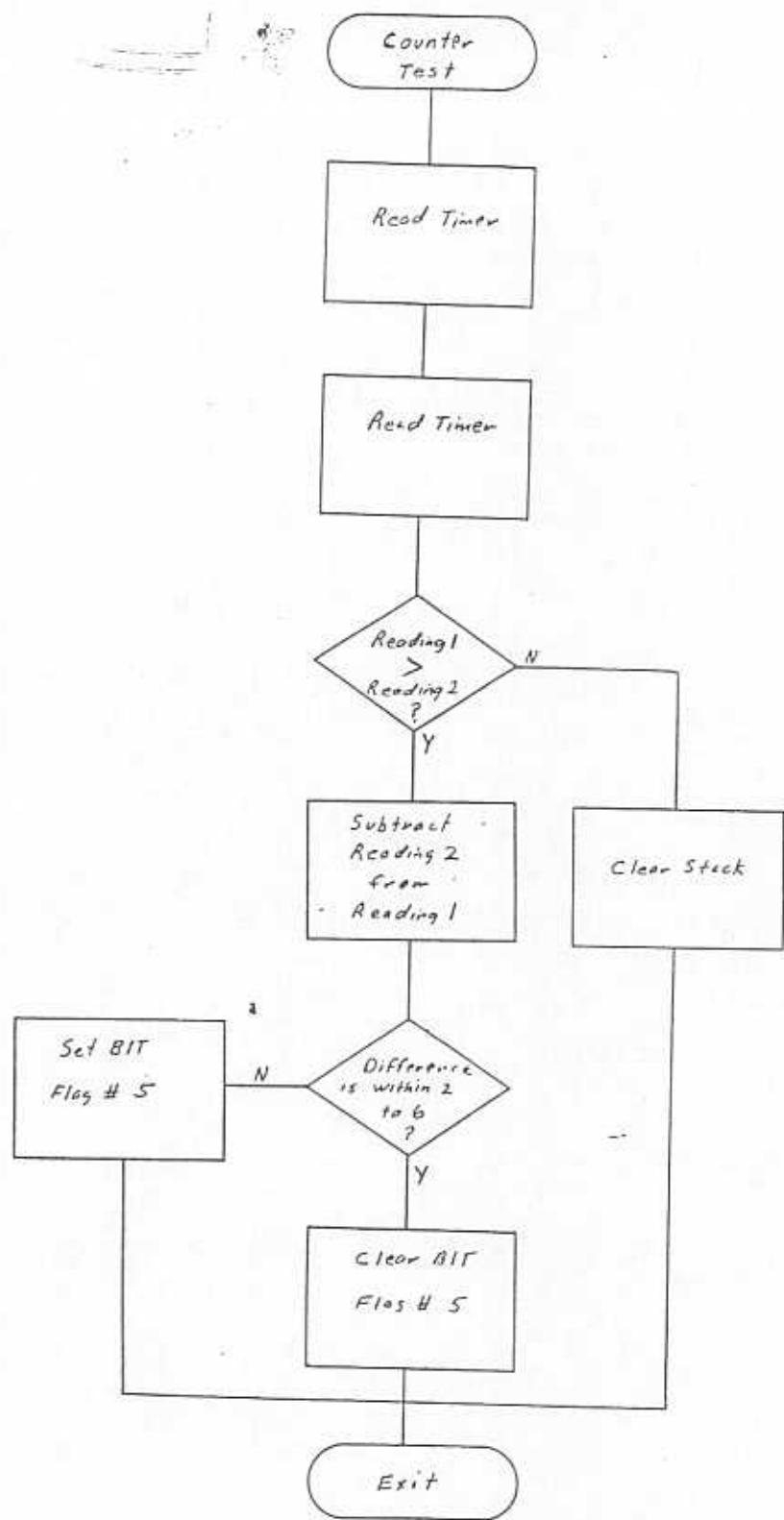


Figure 21

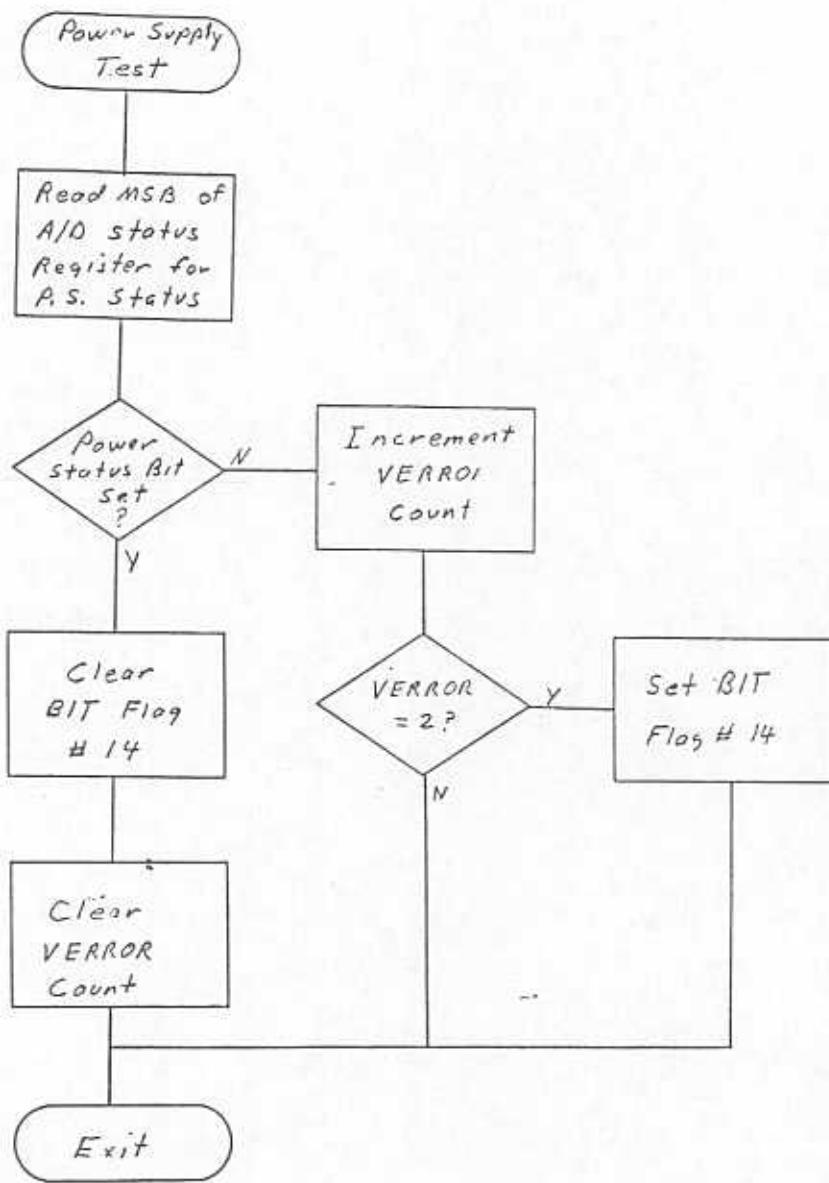


Figure 22

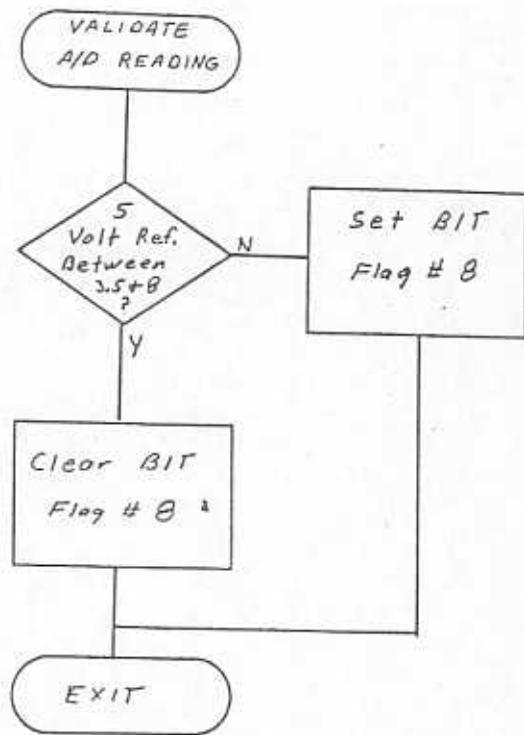


Figure 23

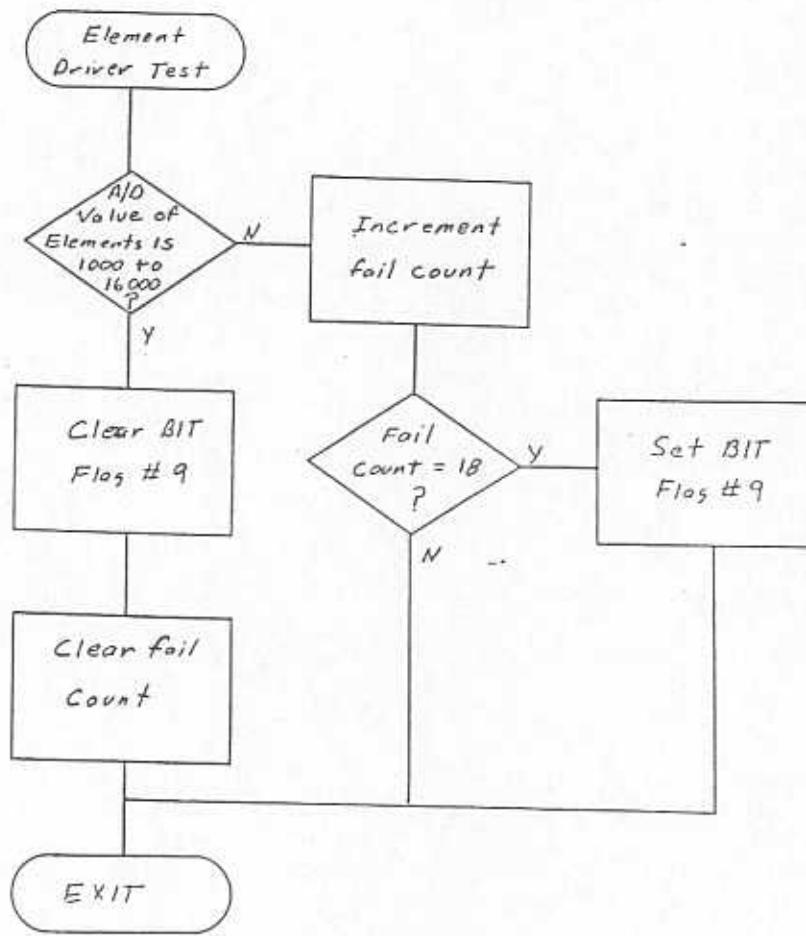


Figure 24

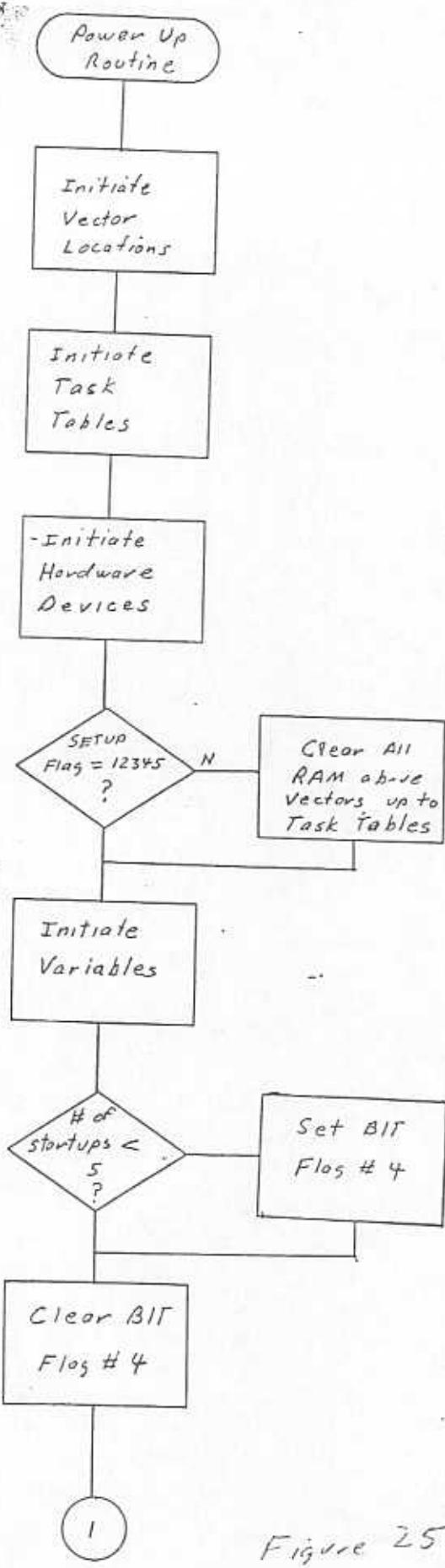


Figure 25

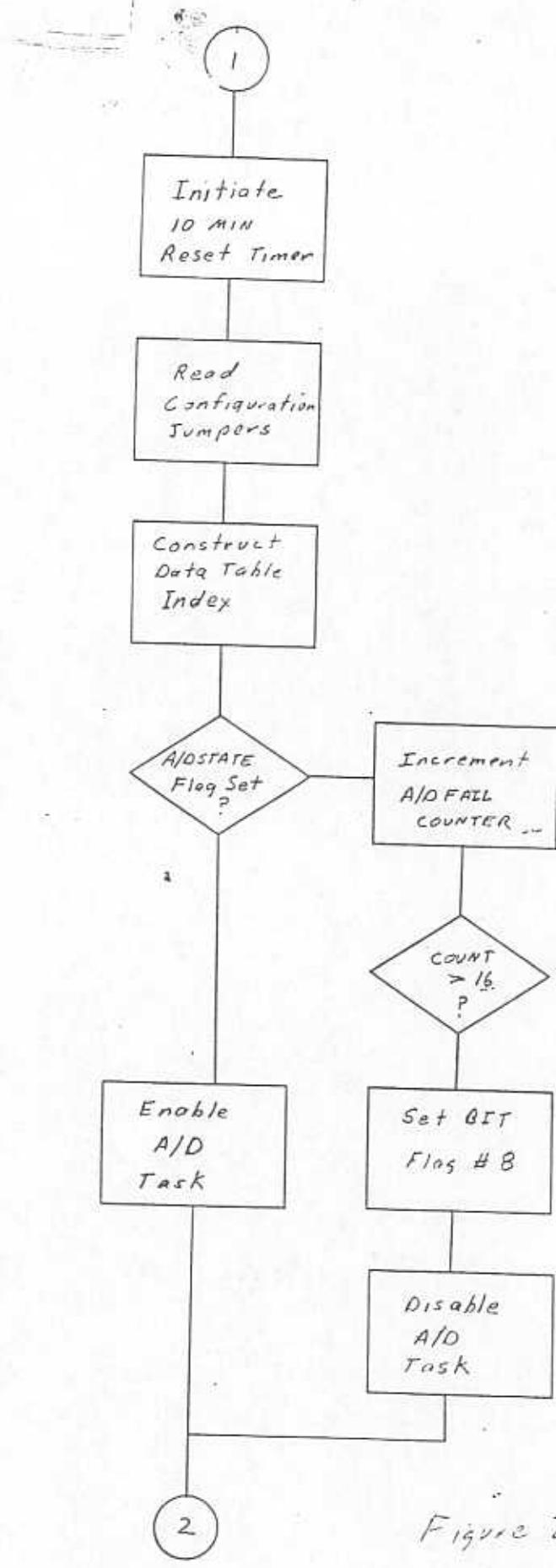


Figure 26

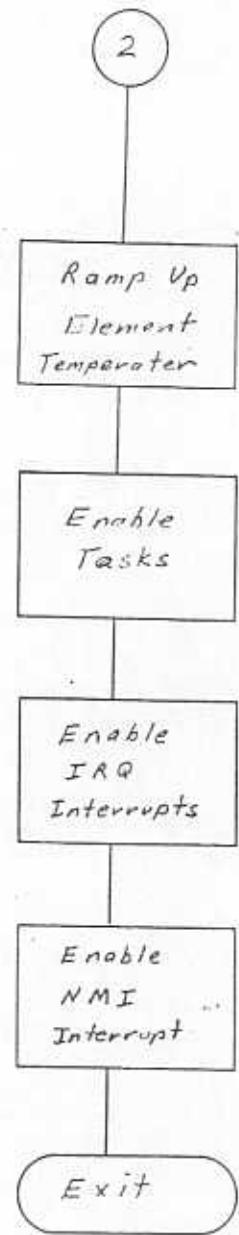


Figure 27

Sensor Characteristics
Data Tables

C000(H)	Serial H's in ASCII	
C000(H)	C060 = CLOCK Freq. C034 = Pressure offset C066 = Speed Index	
C070(H)	ELEMENT RESISTANCE TABLE	106 bytes
C0DA(H)	A/D & D/A GAINS & OFFSETS	22 bytes
COFO(H)	Temperature Lookup Table	
	Pressure Lookup Table	
	X Power Difference Table	
	X Raw Velocity Table	Variable Length
	Y Power Difference Table	
	Y Raw Velocity Table	
	Angle Index	
	Speed Index	
	Magnitude Compensation Table	
	Angle Compensation Table	
	Trig C	

DATA - 2

Figure 29

0000	02 03 D6 AD 50 AD 97 FF 25 0B D6 AC 98 50 AD 97P....%....P..
0010	FF 25 FF	.%.....
0020	FF FF FF FF FF FF FF FF 47 47 4D 4D 89 76 00 8BGGMM.v..
0030	F7 AD 97 FF 25 37 D6 8B 76 00 45 45 AD 97 FF 25%7..v.EE...%
0040	FF
0050	FF
0060	FF
0070	FF
0080	FF
0090	FF
00A0	FF 47 47 57 AD 97 FF 25 4D 4D 89 76 00 5E 47 47	..GGW...%MM.v.^GG
00B0	57 AD 97 FF 25 FF	W...%
00C0	FF FF FF FF 75 02 AD 97 FF 25 FF FF FF FF FF FFu....%
00D0	FF FF FF FF FF FF 8A 45 02 98 03 C3 50 ADE....P.
00E0	97 FF 25 E5 D6 AC 98 03 F0 AD 97 FF 25 EF D6 58	..%.....%..X
00F0	0B C0 74 F1 46 AD 97 FF 25 FB D6 83 ED 04 8F 46	..t.F...%.....F
0100	00 8F 46 02 AD 97 FF 25 0A D7 8B 46 00 40 3B 46	..F....%...F.@;F
0110	02 73 1C 89 46 00 AC 98 03 F0 AD 97 FF 25 20 D7	.S..F.....%..
0120	58 8B D0 03 46 00 8B F8 2B 7E 02 33 D7 78 E4 83	X...F....+~.3.x..
0130	C5 04 46 AD 97 FF 25 39 D7 58 03 46 00 EB CF 41	..F...%9.X.F...A
0140	D7 5A 58 23 C2 50 AD 97 FF 25 4C D7 5A 58 0B C2	.ZX#.P...%L.ZX..
0150	50 AD 97 FF 25 57 D7 5A 58 33 C2 50 AD 97 FF 25	P...%W.ZX3.P...%
0160	62 D7 58 0B C0 B8 00 00 75 01 40 50 AD 97 FF 25	b.X.....u.@P...%
0170	72 D7 5A 58 3B C2 EB ED 7A D7 58 0B C0 B8 00 00	r.ZX;...z.X.....
0180	7E 01 40 50 AD 97 FF 25 8A D7 58 0B C0 B8 00 00	~.@P...%..X.....
0190	79 01 40 50 AD 97 FF 25 9A D7 5A 58 2B C2 EB ED	y.@P...%..ZX+...
01A0	A2 D7 58 5A 2B C2 EB E5 AA D7 5A 58 2B C9 2B C2	..XZ+.....ZX+.+.
01B0	13 C9 51 AD 97 FF 25 B9 D7 5F FF 35 AD 97 FF 25	..Q...%..._5...%
01C0	C2 D7 5F 58 AB AD 97 FF 25 CB D7 5F 58 01 05 AD	.._X.....%..._X...
01D0	97 FF 25 D5 D7 5F 2B C0 8A 05 50 AD 97 FF 25 E1	..%..._+....P...%
01E0	D7 5F 58 AA AD 97 FF 25 EA D7 4D 4D 8F 46 00 AD	._X.....%..MM.F..
01F0	97 FF 25 F5 D7 FF 76 00 45 45 AD 97 FF 25 00 D8	..%...v.EE...%
0200	8B FC FF 35 AD 97 FF 25 0A D8 44 44 AD 97 FF 25	...5...%..DD...%
0210	12 D8 5A 58 52 50 AD 97 FF 25 1C D8 8B FC FF 75	..ZXRP...%.....u
0220	02 AD 97 FF 25 27 D8 5F 5A 58 52 57 50 AD 97 FF%'._ZXRWp...
0230	25 33 D8 FF 76 00 AD 97 FF 25 3C D8 58 8B FC 8B	%3..v....%<.X...
0240	0D 2B C1 75 07 05 01 00 44 44 EB 03 B8 00 00 50	.+..u....DD.....P
0250	AD 97 FF 25 56 D8 5A 58 03 C2 50 AD 97 FF 25 61	...%V.ZX..P...%a
0260	D8 5A 58 2B C2 50 AD 97 FF 25 6C D8 58 40 50 AD	.ZX+.P...%l.X@P.
0270	97 FF 25 75 D8 58 48 50 AD 97 FF 25 7E D8 58 40	..%u.XHP...%-.X@
0280	40 50 AD 97 FF 25 88 D8 58 48 48 50 AD 97 FF 25	@P...%...XHHP...%
0290	92 D8 58 D1 E0 50 AD 97 FF 25 9C D8 58 D1 F8 50	..X..P...%..X..P
02A0	AD 97 FF 25 A6 D8 5F 58 2B D2 F7 F7 52 50 AD 97	...%..._X+....RP..
02B0	FF 25 B4 D8 5F 5A 58 F7 E2 F7 F7 52 50 AD 97 FF	.%..._ZX.....RP...
02C0	25 C3 D8 5F 5A 58 F7 EA F7 FF 50 AD 97 FF 25 D1	%..._ZX.....P...%
02D0	D8 5F 58 F7 E7 50 AD 97 FF 25 DC D8 5F 58 99 F7	._X..P...%..._X..
02E0	FF 50 AD 97 FF 25 E8 D8 5F 2B D2 58 F7 F7 52 AD	.P...%...+_X..R.
02F0	97 FF 25 F5 D8 59 5F 58 E3 11 96 D1 CF D1 C7 73	..%..Y_X.....S
0300	02 A4 49 D1 E9 F3 A5 73 01 A4 96 AD 97 FF 25 F5	..I.....S.....%
0310	D8 13 D9 58 F7 D8 50 AD 97 FF 25 1D D9 58 0B C0	...X..P...%..X..
0320	78 F2 50 AD 97 FF 25 62 D7 2B D9 58 5A 3B C2 79	x.P...%b.+.XZ; .Y

0330	01	92	52	AD	97	FF	25	39	D9	58	5A	3B	D0	EB	F0	41	..R...%9.XZ;...A
0340	D9	5A	59	58	2B	FF	3B	C2	79	05	3B	C1	78	01	47	57	.ZYX+.;.Y.;.x.GW
0350	AD	97	FF	25	C3	D6	00	00	C3	D6	01	00	5E	D9	58	59	...%.....^XY
0360	5F	F3	AA	AD	97	FF	25	69	D9	2B	C0	EB	F2	6F	D9	8B%i.+...o..
0370	C4	50	AD	97	FF	25	78	D9	58	0B	C0	74	01	50	50	AD	.P...%x.X..t.PP.
0380	97	FF	25	85	D9	FF	76	02	AD	97	FF	25	8E	D9	FF	76	..%...v....%...v
0390	04	AD	97	FF	25	97	D9	8B	46	00	89	46	02	AD	97	FF%...F..F....
03A0	25	A3	D9	5F	FF	75	02	FF	35	AD	97	FF	25	AF	D9	5F	%..._u...5...%...-
03B0	58	AB	58	AB	AD	97	FF	25	BA	D9	8B	FC	FF	75	02	FF	X.X....%....u..
03C0	35	AD	97	FF	25	C7	D9	83	C4	04	AD	97	FF	25	D0	D9	5...%.....%..
03D0	5A	5F	58	59	57	52	51	50	AD	97	FF	25	DE	D9	8B	FC	Z_XYWRQP...%
03E0	FF	75	06	FF	75	04	AD	97	FF	25	EC	D9	FF	76	02	FF	.u..u....%...v..
03F0	76	00	83	C5	04	AD	97	FF	25	D8	D6	00	D8	D6	08	D8	v.....%.....
0400	D6	0A	D8	D6	0C	D8	D6	0E	D8	D6	10	D8	D6	12	D8	D6[KK..
0410	14	D8	D6	16	D8	D6	18	C3	D6	FF	D7	5B	4B	4B	C7	07	...g.^]....%/.NN.
0420	2E	EA	8B	67	06	5E	5D	AD	97	FF	25	2F	DA	4E	4E	C7	...UV.g.....o.3.
0430	07	FF	D7	55	56	89	67	06	BF	1B	DA	FF	6F	02	33	DA	(..._.....T.5.P.
0440	28	D6	F9	D9	5F	D8	10	D8	B7	D7	54	D8	35	D6	50	DAG.....
0450	93	5F	8B	1D	C7	07	FF	D7	C7	47	14	00	00	8B	7F	08	...5.]..E.....
0460	83	EF	06	89	35	89	5D	02	C7	45	04	00	00	89	7F	06v._OO.%}._.=
0470	93	E9	C3	FB	76	DA	5F	4F	4F	FF	25	7D	DA	5F	8B	3DX.G....
0480	8B	CF	E3	14	EB	F1	88	DA	58	8F	47	16	0B	C0	7E	08	.G.....%....G
0490	89	47	14	8B	7F	0C	EB	DF	AD	97	FF	25	9E	DA	C7	47	...X...G..G....
04A0	18	00	00	58	F7	D8	89	47	14	8F	47	16	8B	7F	0E	EB	.({...{.5.({...{.5
04B0	C6	28	D6	08	DA	7B	DA	35	D6	28	D6	0B	DA	7B	DA	35	.(.m.X....5....
04C0	D6	28	D6	6D	D9	58	D9	86	DA	08	D8	35	D6	CF	DA	FF%....G....P
04D0	B7	98	EF	AD	97	FF	25	D9	DA	8B	47	08	05	10	00	50	...%...+_...GWP..
04E0	AD	97	FF	25	E6	DA	5F	2B	C0	8A	05	47	57	50	AD	97	.%..._f.
04F0	FF	25	F4	DA	B0	20	E9	66	FE	20	20	20	20	20	20	20	..+X..x....+..P
0500	20	03	DB	2B	FF	58	0B	C0	78	10	BF	08	00	2B	C7	50	...Ry...WGW...%()
0510	BA	F9	DA	52	79	02	03	F8	57	47	57	AD	97	FF	25	285.(.X
0520	D6	01	DB	ED	D6	05	86	DA	E3	D6	F6	35	D6	28	D6	58	...5.7..O....X..
0530	D9	1F	DB	35	D6	37	DB	FF	4F	16	8B	7F	16	58	AA	AD	..%E.X<.r....0P.
0540	97	FF	25	45	DB	58	3C	0A	72	02	04	07	04	30	50	AD	..%(...5.({..
0550	97	FF	25	28	D6	D7	DA	11	DA	C0	D7	35	D6	28	D6	C5-5.5.(..
0560	D9	11	DA	B7	D7	D7	DA	1A	D8	5F	D8	35	D6	28	D6	25-5.5.(.%
0570	D8	88	D7	ED	D6	05	09	D6	2D	35	DB	35	D6	28	D6	10C.5...5
0580	D8	FF	D9	B7	D7	A4	D8	10	D8	43	DB	35	DB	10	D8	35	.(.)...`....5.(.
0590	D6	28	D6	7D	DB	1A	D8	60	D7	ED	D6	F7	35	D6	28	D6T.S....m].5.
05A0	FE	D7	1B	D9	54	D9	53	DB	91	DB	6D	DB	5D	DB	35	D6	(.....-5.({....
05B0	28	D6	9E	DB	86	DA	2D	DB	35	D6	28	D6	B7	D7	B0	DB	5.({...T.S...].%.
05C0	35	D6	28	D6	10	D8	54	D9	53	DB	91	DB	5D	DB	25	D8	..._....5.({.T...
05D0	1A	D8	5F	D8	1F	DB	86	DA	35	D6	28	D6	54	D9	C2	DB	-5.@.F.
05E0	2D	DB	35	D6	E6	DB	8B	7E	00	8A	05	98	40	01	46	00	X..u..?..W...%...
05F0	58	0B	C0	75	03	E9	3F	FA	57	AD	97	FF	25	FF	DB	8B	g.+.P.....%(...
0600	67	08	2B	C0	50	BE	00	00	AD	97	FF	25	28	D6	E4	DB	9.%I..Y ^..O
0610	CD	DA	7C	D8	E4	DA	86	DA	2D	DB	E4	DA	86	DA	B1	DA	..N.....%^.X_
0620	FD	DB	28	D6	58	D9	E4	DB	E4	DA	86	DA	35	D6	30	DC	..(X.....5.0.
0630	96	5F	59	5E	D1	E9	F3	A7	74	07	B9	01	00	73	02	F7	_Y^....t....S..
0640	D9	51	96	AD	97	FF	25	49	DC	96	59	5F	5E	03	F9	4F	..Q.....%I..Y ^..O
0650	03	F1	4E	FD	F3	A4	FC	96	AD	97	FF	25	5E	DC	58	5F	..N.....%^.X_

FMQ-13 Object Code, "Digital Wind Sensor Application Program", CPIN 83M-FMQ13-F002-00A 31 May 1990

0660	57	03	F8	4F	80	3D	20	75	03	48	75	F7	50	AD	97	FF	W..O.= u.Hu.P...
0670	25	73	DC	2B	D2	5F	47	57	8A	05	3C	3A	72	06	3C	41	%s.+._GW..<:r.<A
0680	72	0E	2C	07	2C	30	72	08	3A	47	0A	73	03	98	50	42	r.,.,Or.:G.s..PB
0690	52	AD	97	FF	25	97	DC	5F	5A	58	59	52	F7	67	0A	91	R...%..._ZXYR.g..
06A0	F7	67	0A	03	C7	13	CA	5A	50	51	52	AD	97	FF	25	28	.g.....ZPQR...%(
06B0	D6	71	DC	ED	D6	0B	95	DC	58	D9	11	DA	C9	D7	E3	D6	.q.....X.....
06C0	F0	35	D6	C5	DC	8B	3E	62	05	E9	5E	F9	CE	DC	5A	58	.5.....>b..^...ZX
06D0	8B	FC	01	45	02	11	15	AD	97	FF	25	DD	DC	5A	58	F7	...E.....%...ZX.
06E0	D8	83	D2	00	F7	DA	50	52	AD	97	FF	25	28	D6	DB	DCPR...%(...
06F0	CC	DC	35	D6	28	D6	60	D7	10	D8	60	D7	3F	D7	35	D6	.5.(.^...^.?5.
0700	28	D6	EC	DC	88	D7	10	D8	08	D8	35	D6	28	D6	FE	D7	(.....5.(...
0710	88	D7	ED	D6	02	DB	DC	35	D6	28	D6	EC	DC	F4	DC	355.(....5
0720	D6	28	D6	DC	D9	EC	DC	FE	D7	88	D7	ED	D6	05	CC	DC	.(.....
0730	E3	D6	02	C5	D9	35	D6	28	D6	DC	D9	DC	D9	21	DD	EC5.(....!
0740	DC	CC	DC	35	D6	47	DD	58	99	EB	85	4D	DD	58	5A	F7	...5.G.X...M.XZ.
0750	EA	50	52	AD	97	FF	25	59	DD	5F	5A	58	F7	FF	50	AD	.PR...%Y._ZX..P.
0760	97	FF	25	65	DD	58	5A	F7	E2	50	52	AD	97	FF	25	71	..%e.XZ..PR...%q
0770	DD	5F	5A	58	F7	F7	52	50	AD	97	FF	25	7E	DD	89	76	_ZX..RP...%~..V
0780	FE	89	5E	FC	5B	5E	5F	8B	C7	F7	E3	50	8B	CA	8B	C6	..^.[^...P....
0790	F7	E3	03	C8	83	D2	00	OB	F6	79	02	2B	D3	0B	DB	79Y.+...Y
07A0	04	2B	CF	1B	D6	51	52	8B	5E	FC	8B	76	FE	AD	97	FF	.+...QR.^..v....
07B0	25	B3	DD	5F	5A	58	59	OB	FF	9C	79	02	F7	DF	0B	D2	%..._ZXY...y....
07C0	79	02	03	D7	F7	F7	91	F7	F7	9D	79	07	F7	D8	83	D1	y.....y....
07D0	00	F7	D9	50	51	AD	97	FF	25	28	D6	E8	D7	7C	DD	F3	...PQ...%(... ...
07E0	D7	B1	DD	35	D6	E7	DD	58	59	2B	D2	F7	77	0A	91	F7	...5...XY+..w...
07F0	77	0A	50	51	52	AD	97	FF	25	28	D6	E5	DD	43	DB	35	w.PQR...%(...C.5
0800	DB	35	D6	28	D6	F9	DD	B8	D9	F4	DC	ED	D6	F7	35	D6	.5.(....5.
0810	28	D6	10	D8	1A	D8	0C	DD	53	DB	03	DE	6D	DB	5D	DB	(.....S...m.).
0820	35	D6	28	D6	10	DE	86	DA	2D	DB	35	D6	C3	D6	A0	00	5.(....-5.....
0830	C3	D6	A0	00	C3	D6	A4	00	C3	D6	A5	00	C3	D6	A7	00
0840	C3	D6	A8	00	C3	D6	AA	00	C3	D6	AB	00	C3	D6	AC	00
0850	C3	D6	AD	00	C3	D6	D5	00	C3	D6	DB	00	C3	D6	DD	00
0860	C3	D6	DF	00	C3	D6	E0	00	C3	D6	E1	00	C3	D6	E2	00
0870	C3	D6	E3	00	C3	D6	E4	00	C3	D6	E5	00	C3	D6	E6	00
0880	C3	D6	E7	00	C3	D6	E8	00	C3	D6	E9	00	C3	D6	EA	00
0890	C3	D6	EB	00	C3	D6	EC	00	C3	D6	EE	00	C3	D6	F0	00
08A0	C3	D6	F2	00	C3	D6	F4	00	C3	D6	F6	00	C3	D6	FE	00
08B0	C3	D6	00	01	C3	D6	02	01	C3	D6	04	01	C3	D6	06	01
08C0	C3	D6	0E	01	C3	D6	16	01	C3	D6	18	01	C3	D6	1A	01
08D0	C3	D6	3E	01	C3	D6	40	01	C3	D6	42	01	C3	D6	44	01	..>...@...B...D.
08E0	C3	D6	54	01	C3	D6	5C	01	C3	D6	64	01	C3	D6	78	01	.T...\\...d...x.
08F0	C3	D6	8C	01	C3	D6	8E	01	C3	D6	90	01	C3	D6	92	01
0900	C3	D6	94	01	C3	D6	96	01	C3	D6	98	01	C3	D6	9A	01
0910	C3	D6	9C	01	C3	D6	A0	01	C3	D6	A4	01	C3	D6	A6	01
0920	C3	D6	A8	01	C3	D6	A9	01	C3	D6	AA	01	C3	D6	AB	01
0930	C3	D6	AC	01	C3	D6	AD	01	C3	D6	AE	01	C3	D6	AF	01
0940	C3	D6	B0	01	C3	D6	B1	01	C3	D6	B2	01	C3	D6	B3	01
0950	C3	D6	B4	01	C3	D6	B5	01	C3	D6	B7	01	C3	D6	B9	01
0960	C3	D6	BB	01	C3	D6	BD	01	C3	D6	BF	01	C3	D6	C1	01
0970	C3	D6	C3	01	C3	D6	C5	01	C3	D6	C7	01	C3	D6	C9	01
0980	C3	D6	CB	01	C3	D6	CD	01	C3	D6	CF	01	C3	D6	D1	01

FMQ-13 Object Code, "Digital Wind Sensor Application Program". CPIN 83M-FMQ13-F002-00A 31 May 1990

0990	C3 D6 D3 01 C3 D6 D5 01 C3 D6 D7 01 C3 D6 D9 01	
09A0	C3 D6 DD 01 C3 D6 E1 01 C3 D6 E5 01 C3 D6 E9 01
09B0	C3 D6 ED 01 C3 D6 F1 01 C3 D6 F2 01 C3 D6 F6 01
09C0	C3 D6 F8 01 C3 D6 FC 01 C3 D6 FE 01 C3 D6 FF 01
09D0	C3 D6 00 02 C3 D6 01 02 C3 D6 02 02 C3 D6 03 02
09E0	C3 D6 04 02 C3 D6 06 02 C3 D6 08 02 C3 D6 0A 02
09F0	C3 D6 0C 02 C3 D6 10 02 C3 D6 14 02 C3 D6 18 02
0A00	C3 D6 1C 02 C3 D6 1E 02 C3 D6 20 02 C3 D6 24 02
0A10	C3 D6 28 02 C3 D6 2C 02 C3 D6 30 02 C3 D6 34 02\$.
0A20	C3 D6 36 02 C3 D6 00 C0 C3 D6 60 C0 C3 D6 64 C0	..(....0...4.
0A30	C3 D6 66 C0 C3 D6 68 C0 C3 D6 6A C0 C3 D6 6C C0	..6.....`...d.
0A40	C3 D6 6E C0 C3 D6 00 08 C3 D6 00 14 C3 D6 00 80	..f...h...j...l.
0A50	C3 D6 01 80 C3 D6 02 80 C3 D6 00 60 C3 D6 01 60	..n.....
0A60	C3 D6 02 60 C3 D6 03 60 C3 D6 00 00 C3 D6 01 00`....
0A70	C3 D6 40 00 C3 D6 00 40 C3 D6 01 40 C3 D6 02 40	...@...@...@...@
0A80	C3 D6 03 40 C3 D6 50 00 A1 D6 0E 00 C0 EB 29 F5	...@...P.....).
0A90	D5 FE D0 07 39 0F 0B 15 A6 19 9C E0 90 AD 97 FF9.....
0AA0	25 A3 E0 90 AD 97 FF 25 AA E0 90 AD 97 FF 25 B1	%.....%.....%
0AB0	E0 90 AD 97 FF 25 B8 E0 90 AD 97 FF 25 BF E0 90%.....%
0AC0	AD 97 FF 25 C6 E0 90 AD 97 FF 25 CD E0 58 59 5A	...%.....%..XYZ
0AD0	52 51 50 AD 97 FF 25 D9 E0 58 59 5A 52 51 50 AD	RQP...%..XYZRQP.
0AE0	97 FF 25 E5 E0 58 59 5A 52 51 50 AD 97 FF 25 28	..%..XYZRQP...%(
0AF0	D6 09 D6 10 FF D9 C0 D7 35 D6 28 D6 09 D6 0A FF5.(....
0B00	D9 C0 D7 35 D6 28 D6 1A D8 54 D8 10 D8 35 D6 28	...5.(...T...5.(....
0B10	D6 B7 D7 90 D8 54 D8 B7 D7 35 D6 28 D6 E8 D7 94T...5.(....
0B20	DE B7 D7 1A D8 31 D8 94 DE 0F E1 98 D7 27 D9 ED1.....'..
0B30	D6 05 98 DE E3 D6 02 9C DE C0 D7 98 DE B7 D7 9C
0B40	DE B7 D7 1A D8 5F D8 FE D7 E8 D7 9A D8 54 D8 94_.....T..
0B50	DE C0 D7 F3 D7 09 D6 02 98 D7 2B DA ED D6 C0 F3+.....
0B60	D7 08 D8 35 D6 28 D6 FE D7 B7 D7 9A D8 FE D7 9C	...5.(.....
0B70	DE C0 D7 9A D8 94 DE C0 D7 58 D9 98 DE C0 D7 35X.....5
0B80	D6 83 E1 58 08 06 EB 00 8A 06 EB 00 88 06 00 80	...X.....
0B90	AD 97 FF 25 96 E1 58 34 FF 20 06 EB 00 8A 06 EB	...%..X4.
0BA0	00 88 06 00 80 AD 97 FF 25 AB E1 5F 5A 58 F7 F7%..ZX..
0BB0	50 AD 97 FF 25 B7 E1 59 58 05 01 00 83 D1 00 D1	P...%..YX.....
0BC0	F9 D1 D8 50 AD 97 FF 25 CA E1 59 58 5A D1 F8 D1	...P...%..YXZ...
0BD0	DA E2 FA 52 AD 97 FF 25 DA E1 59 58 D1 E8 E2 FC	...R...%..YX....
0BE0	50 AD 97 FF 25 E7 E1 59 BF A0 00 85 4D 02 75 05	P...%..Y...M.u.
0BF0	83 F1 FF 21 0D AD 97 FF 25 FB E1 59 BF A0 00 85	...!....%..Y....
0C00	0D 75 05 09 0D 09 4D 02 AD 97 FF 25 28 D6 54 D9	.u....M....%(.T.
0C10	1C DF B7 D7 01 D6 C4 09 A0 D7 ED D6 05 27 D9 E3'..
0C20	D6 06 58 D9 1C DF C9 D7 35 D6 2C E2 B8 55 55 8B	..X....5.,,UU.
0C30	C8 BA AA AA 03 C2 79 04 33 C1 D1 E8 2B C8 75 04y.3...+..u.
0C40	D1 E2 2B C2 0B C1 50 AD 97 FF 25 4D E2 8A 06 B2	..+...P...%M....
0C50	01 E6 50 AD 97 FF 25 28 D6 4C DF FE D7 D3 D7 6A	..P...%(.L....j
0C60	D8 09 D6 02 29 D9 FE D7 25 D8 DF D7 35 D6 28 D6)....%....5.(.
0C70	58 D9 2A E2 58 D9 70 D7 ED D6 05 E5 E1 E3 D6 02	X.*.X.p.....
0C80	F9 E1 4B E2 2B DA 35 D6 8A E2 BA 00 00 B9 08 00	..K.+.5.....
0C90	5F FA 8A 05 C6 05 55 80 3D 55 75 0B C6 05 AA 80U.=Uu.....
0CA0	3D AA 74 01 42 EB 01 42 88 05 47 FB E2 E3 52 AD	=t.B..B..G...R.
0CB0	97 FF 25 28 D6 54 D9 38 DF DF D7 09 D6 04 54 D9	..%(.T.8.....T.

0CC0	01 D6 00 08 54 D9 F9 D6 2B DA 4B E2 38 DF D3 D7T....+.K.8...
0CD0	0C E2 4A D7 ED D6 F1 54 D9 38 DF DF D7 31 D8 88	..J....T.8...1...
0CE0	E2 FE D7 ED D6 02 95 D9 54 D8 4B E2 2B DA 09 D6T.K.+...
0CF0	08 1E D7 D4 ED D6 05 F9 E1 E3 D6 02 E5 E1 4B E2K.
0D00	2B DA 35 D6 06 E3 8B FE 5E 5A 2B C0 B9 08 00 AC	+.5.....^Z+....
0D10	03 D0 E2 FB 8B F7 52 AD 97 FF 25 1D E3 2B C0 8AR...%...+..
0D20	06 02 80 24 80 50 AD 97 FF 25 28 D6 09 D6 02 54	...\$.P...%(....T
0D30	D9 09 D6 F8 01 D6 00 C0 F9 D6 31 D8 04 E3 4B E21...K.
0D40	2B DA 09 D6 08 1E D7 F2 09 D6 FE B7 D7 70 D7 ED	+.....p..
0D50	D6 07 E5 E1 A1 E0 E3 D6 02 F9 E1 4B E2 2B DA 35K.+.5
0D60	D6 63 E3 FA C6 06 03 60 00 8A 0E 00 60 8A 06 00	.c.....`.....
0D70	60 8A E8 51 B9 30 00 E2 FE C6 06 03 60 00 8A 0E	`..Q.0.....`..
0D80	00 60 8A 06 00 60 8A E8 FB 51 AD 97 FF 25 28 D6	..`....`...Q...%(.
0D90	54 D9 38 DF DF D7 2B DA 4B E2 38 DF D3 D7 0C E2	T.8...+.K.8.....
0DA0	4A D7 ED D6 F1 09 D6 20 61 E3 B8 D9 A8 D7 27 D9	J.....a.....!
0DB0	ED D6 1D 5F D8 09 D6 02 09 D6 06 3F D9 0C E2 27	..._.....?....
0DC0	D9 3F D7 ED D6 05 E5 E1 E3 D6 02 F9 E1 E3 D6 04	.?.....
0DD0	C5 D9 08 D8 4B E2 2B DA 35 D6 28 D6 01 D6 00 40K.+.5. (....@
0DE0	1B E3 ED D6 0B E5 E1 54 D9 4C DF DF D7 E3 D6 11T.L.....
0DF0	57 E2 09 D6 02 70 D7 ED D6 05 F9 E1 E3 D6 02 08	W....p.....
0E00	D8 4B E2 2B DA 35 D6 28 D6 4B E2 2B DA 2A E3 8E	.K.+.5. (.K.+.*..
0E10	E3 DA E3 B3 E2 6E E2 E3 D6 EF 35 D6 1E E4 C6 06n....5.....
0E20	E5 00 00 B9 04 00 58 BA 01 00 02 16 E5 00 EE 8AX.....
0E30	C4 42 EE 80 06 E5 00 10 E2 EC 88 06 B2 01 AD 97	.B.....
0E40	FF 25 44 E4 8A 06 B2 01 E6 40 C6 06 B1 01 00 AD	.%D.....@.....
0E50	97 FF 25 55 E4 80 26 EB 00 F8 8A 06 EB 00 88 06	..%U..&.....
0E60	00 80 C6 06 E8 00 01 C6 06 E9 00 00 B9 04 00 BF
0E70	F6 00 C7 05 00 00 47 47 E2 F8 C6 06 02 80 00 8AGG.....
0E80	06 EB 00 0A 06 E8 00 88 06 00 80 C6 06 A9 01 0F
0E90	9B C6 06 A9 01 00 C6 06 A8 01 00 2B C0 8A 06 00+.....
0EA0	80 8A E0 8A 06 01 80 C6 06 02 80 00 B9 04 00 D3
0EB0	E8 8A 0E EA 00 D1 E1 8B F9 01 85 F6 00 FE 06 EA
0EC0	00 80 26 EA 00 03 F6 06 E8 00 03 75 0F FE 06 E9	..&.....u....
0ED0	00 F6 06 E9 00 03 75 04 AD 97 FF 25 FE 06 E8 00u....%
0EE0	80 26 E8 00 03 8A 06 EB 00 0A 06 E8 00 88 06 00	.&.....
0EF0	80 C6 06 A9 01 0F EB 98 FA E4 8A 06 EB 00 59 0AY.....
0F00	C1 88 06 00 80 B9 05 00 E2 FE 88 06 02 80 2B C0+.....
0F10	C6 06 A9 01 0F 9B C6 06 A9 01 00 C6 06 A8 01 00
0F20	8A 06 00 80 8A E0 8A 06 01 80 B9 04 00 D3 E8 50P
0F30	AD 97 FF 25 36 E5 8A 06 B4 01 0A C0 75 3E C6 06	...%6.....u>...
0F40	03 60 E4 8A 06 01 60 A8 80 75 2F C6 06 03 60 D8	..`....`..u/...`..
0F50	8B 06 9A 01 89 06 00 01 8A 06 02 60 8A 26 02 60`..&..
0F60	F7 D8 89 06 9A 01 C6 06 03 60 B0 2B C0 88 06 02`..+....
0F70	60 88 06 02 60 C6 06 B4 01 01 EB 12 C6 06 03 60	~...`.....`..
0F80	E4 8A 06 01 60 A8 80 74 05 C6 06 B4 01 00 AD 97`..t.....
0F90	FF 25 28 D6 09 D6 04 F8 E4 B4 DE C9 D7 09 D6 05	.%{.....
0FA0	F8 E4 04 DF C9 D7 09 D6 06 F8 E4 FE D7 B8 DE C9
0FB0	D7 F8 DE C0 D7 09 D6 07 F8 E4 00 DF C9 D7 35 D65.
0FC0	28 D6 01 D6 00 01 F8 DE B7 D7 01 D6 99 05 01 D6	(.....
0FD0	CD 0C 3F D9 ED D6 05 E5 E1 E3 D6 02 F9 E1 35 D6	..?.....5.
0FE0	28 D6 2B DA 30 DF D3 D7 ED D6 1B 54 D9 30 DF DF	(.+.0.....T.0..

FMQ-13 Object Code. "Digital Wind Sensor Application Program". CPIN 83M-FMQ13-F002-00A 31 May 1990

0FF0	D7	53	E4	92	E5	44	DF	D3	D7	ED	D6	02	42	E4	C0	E5	.S...D.....B...
1000	58	D9	80	DE	DF	D7	E3	D6	D9	35	D6	0D	E6	83	EC	08	X.....5.....
1010	8B	FC	8B	45	0E	8B	4D	0A	F7	E1	89	55	04	89	45	06	..E..M....U..E.
1020	31	C0	89	45	02	89	05	8B	45	0C	F7	E1	01	45	04	11	1..E....E....E..
1030	55	02	83	15	00	8B	4D	08	8B	45	0E	F7	E1	01	45	04	U.....M..E....E.
1040	11	55	02	83	15	00	8B	45	0C	F7	E1	01	45	02	11	15	.U.....E....E...
1050	FA	83	C4	10	FF	75	06	FF	75	04	FF	75	02	FF	35	FBu..u..u..5.
1060	AD	97	FF	25	66	E6	8B	FC	8B	45	06	01	45	0E	8B	45	...%f....E..E..E
1070	04	11	45	0C	8B	45	02	11	45	0A	8B	05	11	45	08	83	..E..E..E....E..
1080	C4	08	AD	97	FF	25	C3	D6	00	40	8C	E6	58	59	F7	E9%....@..XY..
1090	D1	E0	D1	D2	D1	E0	D1	D2	52	AD	97	FF	25	9F	E6	59R...%..Y
10A0	5A	2B	C0	D1	FA	D1	D8	D1	FA	D1	D8	F7	F9	50	AD	97	Z+.....P..
10B0	FF	25	B4	E6	5F	58	2B	D2	F7	F7	50	AD	97	FF	25	C1	.%.._X+...P...%.
10C0	E6	58	D1	E0	D1	E0	79	02	F7	D8	2D	00	40	F7	D8	50	.X....Y....@..P
10D0	AD	97	FF	25	D6	E6	58	2D	00	10	50	AD	97	FF	25	28	...%..X-..P...%(
10E0	D6	BF	E6	FE	D7	FE	D7	8A	E6	FE	D7	09	D6	B9	8A	E6
10F0	01	D6	15	05	54	D8	1A	D8	8A	E6	01	D6	AA	D6	54	D8T.....T.
1100	8A	E6	01	D6	88	24	54	D8	1A	D8	8A	E6	54	D8	2B	DA\$T.....T.+.
1110	35	D6	28	D6	D4	E6	DF	E6	35	D6	1C	E7	58	D1	C8	72	5.(.....5...X..r
1120	01	AD	D1	56	00	AD	97	FF	25	00	F4	00	04	00	34	00	...V....%.....4.
1130	C4	00	14	00	E4	00	D4	00	24	28	D6	FE	D7	88	D7	E8\$(..
1140	D7	1B	D9	10	D8	FE	D7	88	D7	1A	E7	11	D9	B8	D9	A0
1150	D7	1A	E7	10	D8	9D	E6	FE	D7	01	D6	90	E5	54	D8	10
1160	D8	01	D6	82	1A	8A	E6	86	E6	54	D8	9D	E6	FE	D7	FET.....
1170	D7	8A	E6	FE	D7	01	D6	B6	01	8A	E6	01	D6	A0	FC	54T.....
1180	D8	8A	E6	01	D6	2F	0A	54	D8	8A	E6	F3	D7	90	D8	01/..T.....
1190	D6	29	E7	54	D8	B7	D7	54	D8	1B	D9	2B	DA	35	D6	28).).T...T...+.5.(
11A0	D6	01	D6	68	01	9D	E6	35	D6	28	D6	54	D9	01	D6	50	...h...5.(.T...P
11B0	46	01	D6	00	20	D9	DD	08	D8	35	D6	28	D6	01	D6	00	F...5.(....
11C0	20	1A	D8	98	D7	ED	D6	04	86	E6	5F	D8	35	D6	D0	E75...
11D0	5A	58	56	2B	FF	8B	F7	B9	10	00	D1	E0	D1	D2	D1	D6	ZXV+.....
11E0	D1	E0	D1	D2	D1	D6	D1	E7	D1	E7	47	3B	F7	72	03	2BG;..r.+
11F0	F7	47	D1	EF	E2	E4	5E	57	AD	97	FF	25	FE	E7	83	EC	.G....^W...%....
1200	04	8B	FC	31	D2	31	C0	89	05	89	45	02	B9	20	00	D1	..1.1....E....
1210	65	0A	D1	55	08	D1	55	06	D1	55	04	D1	D0	D1	D2	D1	e..U..U..U.....
1220	65	0A	D1	55	08	D1	55	06	D1	55	04	D1	D0	D1	D2	D1	e..U..U..U.....
1230	25	D1	55	02	D1	25	D1	55	02	83	05	01	83	55	02	00	%..U..%..U.....U..
1240	3B	55	02	72	12	75	04	3B	05	72	0C	2B	05	1B	55	02	;U.r.u.;.r.+..U..
1250	83	05	01	83	55	02	00	D1	6D	02	D1	1D	E2	B1	8B	05U...m.....
1260	8B	55	02	83	C4	0C	50	52	AD	97	FF	25	28	D6	34	DE	.U....PR...%(..4.
1270	D3	D7	09	D6	20	3F	D7	ED	D6	06	01	D6	80	00	94	E1?
1280	35	D6	28	D6	01	D6	80	00	AC	DE	B7	D7	C8	DF	D3	D7	5.(.....
1290	ED	D6	1D	01	D6	68	F7	01	D6	DC	05	3F	D9	ED	D6	05h....?
12A0	81	E1	E3	D6	08	94	E1	54	D9	C8	DF	DF	D7	E3	D6	1AT.....
12B0	01	D6	30	F8	01	D6	E8	03	3F	D9	ED	D6	0B	81	E1	58	..0....?
12C0	D9	C8	DF	DF	D7	E3	D6	02	08	D8	35	D6	28	D6	5F	D85.(..
12D0	4B	DD	01	D6	00	20	54	D9	CC	DC	09	D6	0E	C8	E1	35	K... T.....5
12E0	D6	28	D6	4B	DD	01	D6	00	40	54	D9	CC	DC	09	D6	0D	.(.K....@T.....
12F0	C8	E1	54	D8	09	D6	02	D8	E1	35	D6	28	D6	98	DE	B7	..T.....5.(....
1300	D7	90	D8	5C	DF	B7	D7	54	D8	FE	D7	E8	D7	B7	D7	5F	..\...T.....
1310	D8	01	D6	18	FC	F3	D7	A1	D9	B8	D9	10	E0	AD	D9	5F

1320	D8	C1	D8	09	D6	08	98	DE	B7	D7	5F	D8	01	D6	E8	03
1330	CF	D8	54	D8	35	D6	28	D6	58	DF	B7	D7	09	D6	10	54	.T.5.(.X.....T
1340	D8	A1	D9	25	D8	10	D8	CC	E8	35	D6	28	D6	B4	DE	B7%.....5.(....
1350	D7	36	E9	7C	D8	09	D6	04	DA	D8	B8	DE	B7	D7	36	E9	.6.6.
1360	7C	D8	09	D6	04	DA	D8	B4	DE	A1	D9	CC	DE	09	D6	0C
1370	54	D8	AD	D9	FE	D7	FC	DE	C0	D7	54	D9	B4	DE	C0	D7	T.....T.....
1380	54	D9	B8	DE	C0	D7	35	D6	28	D6	D0	DE	B7	D7	54	D8	T.....5.(....T.
1390	35	D6	28	D6	30	DE	B7	D7	01	D6	00	01	3F	D7	27	D9	5.(.0.....?.'.
13A0	ED	D6	32	01	D6	00	04	4B	E9	44	E0	10	D8	C1	D8	5C	.2....K.D.....\
13B0	DF	B7	D7	65	E1	1B	E1	FB	E8	FE	D7	01	D6	84	EA	01	...e.....
13C0	D6	B0	1D	3F	D9	ED	D6	09	AC	DE	C0	D7	E5	E1	E3	D6	...?.....
13D0	04	08	D8	F9	E1	35	D6	28	D6	09	D6	1A	CF	D8	54	DF5.(....T.
13E0	B7	D7	54	D8	35	D6	28	D6	FE	D7	FE	D7	01	D6	88	13	.T.5.(.....
13F0	01	D6	68	42	3F	D9	ED	D6	12	01	D6	E8	03	A4	D8	09	.hb?.....
1400	D6	05	5F	D8	54	D9	10	D8	E3	D6	35	FE	D7	01	D6	88	.._T.....5.....
1410	13	98	D7	ED	D6	1F	FE	D7	88	D7	ED	D6	09	54	D9	58T.X
1420	D9	5F	D8	E3	D6	02	54	D9	01	D6	88	13	54	D9	EC	DCT.....T...
1430	54	D9	E3	D6	0B	01	D6	80	3E	5F	D8	54	D9	09	D6	0B	T.....>_T....
1440	35	D6	28	D6	FE	D7	E8	D7	D7	E9	E8	D7	E6	E9	90	D8	5.(.....
1450	F3	D7	54	D8	A1	D9	B8	D9	14	E0	AD	D9	FE	D7	E8	D7	..T.....
1460	5F	D8	01	D6	E8	03	D9	DD	F3	D7	54	D9	CC	DC	01	D6T.....
1470	50	C3	54	D9	CC	DC	B8	D9	DC	DE	31	D8	09	D6	04	CF	P.T.....1.....
1480	D8	54	D8	AD	D9	01	D6	00	10	01	D6	40	1F	D9	DD	09	.T.....@....
1490	D6	03	54	D9	CC	DC	09	D6	05	57	DD	31	D8	09	D6	04	.T.....W.1....
14A0	CF	D8	58	DF	B7	D7	54	D8	A1	D9	25	D8	10	D8	E1	E8	.X...T...%.....
14B0	F3	D7	90	D8	BC	DE	54	D8	C0	D7	35	D6	28	D6	AC	DET...5.(...
14C0	B7	D7	88	E9	09	D6	04	54	D9	F9	D6	31	D8	42	EA	2BT...1.B.+
14D0	DA	08	D7	F7	D4	DE	C0	D7	35	D6	28	D6	BC	DE	FE	D75.(.....
14E0	E8	D7	09	D6	06	54	D8	B7	D7	31	D8	09	D6	04	54	D8T...1....T.
14F0	B7	D7	31	D8	7C	D8	B7	D7	F3	D7	B7	D7	35	D6	28	D6	.1.5.(.
1500	92	E9	BC	EA	DA	EA	1C	E4	82	E8	35	D6	28	D6	C4	DF5.(...
1510	B7	D7	64	DF	B7	D7	65	E1	1B	E1	98	DE	B7	D7	FE	D7	..d...e.....
1520	98	DF	C0	D7	90	D8	64	DF	B7	D7	54	D8	A1	D9	A0	DFd...T.....
1530	AD	D9	35	D6	28	D6	BC	DF	B7	D7	68	DF	B7	D7	65	E1	.5.(.....h...e.
1540	1B	E1	98	DE	B7	D7	FE	D7	94	DF	C0	D7	90	D8	68	DFh.....h.
1550	B7	D7	54	D8	A1	D9	9C	DF	AD	D9	35	D6	28	D6	94	DF	..T.....5.(...
1560	B7	D7	73	D8	64	DF	B7	D7	B7	D7	FE	D7	E8	D7	CF	D8	..s.d.....
1570	98	DF	B7	D7	73	D8	90	D8	54	D8	54	D8	FE	D7	A1	D9s...T.T.....
1580	A4	DF	AD	D9	F3	D7	54	D8	A1	D9	A8	DF	AD	D9	35	D6T.....5.
1590	28	D6	E8	D7	A4	DF	B7	D7	10	D8	A0	DF	B7	D7	5F	D8	(.....___.-
15A0	FE	D7	E8	D7	A4	DF	A1	D9	5F	D8	A0	DF	A1	D9	5F	D8___.-
15B0	FE	D7	E8	D7	C1	D8	54	D8	FE	D7	A8	DF	FE	D7	B7	D7T.....
15C0	10	D8	A1	D9	5F	D8	EA	D9	C1	D8	54	D8	10	D8	5F	D8___.-
15D0	F3	D7	9C	DF	B7	D7	5F	D8	9C	DF	A1	D9	5F	D8	C1	D8___.-
15E0	54	D8	35	D6	28	D6	04	DF	B7	D7	36	E9	00	DF	B7	D7	T.5.(.....6....
15F0	36	E9	54	D9	04	DF	C0	D7	54	D9	00	DF	C0	D7	35	D6	6.T.....T.....5.
1600	28	D6	FE	D7	E8	D7	A1	D9	25	D8	45	DD	F3	D7	AD	D9	(.....%..E....
1610	35	D6	28	D6	E4	EB	44	E0	FC	DE	B7	D7	C1	D8	B8	DF	5.(...D.....
1620	00	EC	44	E0	FC	DE	B7	D7	C1	D8	C0	DF	00	EC	35	D6	..D.....5.
1630	28	D6	01	D6	00	20	0C	EB	34	EB	6C	DF	B7	D7	5C	EB	(.....4.1...\\.
1640	90	EB	FE	D7	01	D6	7C	15	01	D6	04	29	3F	D9	ED	D6)?...

1650	09 D8 DE C0 D7 E5 E1 E3 D6 04 08 D8 F9 E1 35 D65.
1660	28 D6 28 DF D3 D7 76 D9 ED D6 05 73 D8 E3 D6 45	(.v....s...E
1670	09 D6 07 B8 DF A1 D9 B8 D9 AC DF AD D9 09 D6 04
1680	45 DD 09 D6 03 C8 E1 BC DF C0 D7 C0 DF A1 D9 B8	E.....
1690	D9 B0 DF AD D9 09 D6 04 45 DD 09 D6 03 C8 E1 C4E.....
16A0	DF C0 D7 54 D9 54 D9 B8 DF AD D9 54 D9 54 D9 C0T.T.....T.T..
16B0	DF AD D9 30 EC 28 DF DF D7 35 D6 28 D6 30 DE B7	...0.(...5.(.0..
16C0	D7 01 D6 00 01 3F D7 27 D9 34 DF D3 D7 B4 DF DF?.'4.....
16D0	D7 ED D6 22 CC DF D3 D7 ED D6 05 12 EC E3 D6 14".....
16E0	01 D6 F0 00 CC DF DF D7 54 D9 04 DF C0 D7 54 D9T.....T.
16F0	00 DF C0 D7 60 EC 35 D6 28 D6 F4 DE B7 D7 84 DF`5.(...
1700	B7 D7 65 E1 1B E1 98 DE B7 D7 FE D7 E8 DF C0 D7	..e.....
1710	90 D8 84 DF B7 D7 54 D8 A1 D9 F0 DF AD D9 35 D6T.....5.
1720	28 D6 F0 DE B7 D7 88 DF B7 D7 65 E1 1B E1 98 DE	(.....e.....
1730	B7 D7 FE D7 EC DF C0 D7 90 D8 88 DF B7 D7 54 D8T.....T.
1740	A1 D9 F4 DF AD D9 35 D6 28 D6 EC DF B7 D7 73 D85.(...s.
1750	84 DF B7 D7 B7 D7 FE D7 E8 D7 CF D8 E8 DF B7 D7
1760	73 D8 90 D8 54 D8 54 D8 FE D7 A1 D9 F8 DF AD D9	s...T.T.....
1770	F3 D7 54 D8 A1 D9 FC DF AD D9 35 D6 28 D6 E8 D7	..T.....5.(...
1780	F8 DF B7 D7 10 D8 F0 DF B7 D7 5F D8 FE D7 E8 D7
1790	F8 DF A1 D9 5F D8 F0 DF A1 D9 5F D8 FE D7 E8 D7
17A0	C1 D8 54 D8 FE D7 FC DF FE D7 B7 D7 10 D8 A1 D9	..T.....
17B0	5F D8 EA D9 C1 D8 54 D8 10 D8 5F D8 F3 D7 F4 DFT.....
17C0	B7 D7 5F D8 F4 DF A1 D9 5F D8 C1 D8 54 D8 35 D6T.....T.5.
17D0	28 D6 40 DF FE D7 D3 D7 6A D8 09 D6 05 E6 D8 10	(. @....j.....
17E0	D8 DF D7 35 D6 28 D6 B8 D9 1C E0 AD D9 E8 DE 40	...5.(...@....@
17F0	DF D3 D7 09 D6 04 CF D8 FE D7 E8 D7 54 D8 AD D9T...
1800	B8 D9 20 E0 AD D9 EC DE F3 D7 54 D8 AD D9 D0 EDT.....
1810	35 D6 28 D6 F9 D6 0C DD B8 D9 0B E6 EA D9 0C DD	5.(...
1820	B8 D9 0B E6 64 E6 FC E7 2B DA 35 D6 28 D6 FE D7d...+5.(...
1830	4B DD 25 D8 FE D7 4B DD CC DC CE E7 35 D6 28 D6	K.%...K....5.(...
1840	B8 D9 F9 D6 CE D9 B8 D9 F9 D6 12 EE 09 D6 02 54T
1850	D9 CC DC 09 D6 04 57 DD EA D9 09 D6 04 57 DD EAW.....W..
1860	D9 09 D6 04 57 DD 10 D8 39 E7 F4 DE C0 D7 F0 DEW...9.....
1870	C0 D7 35 D6 28 D6 B8 D9 DF E6 4B DD 58 D9 7C DD	.5.(...K.X. .
1880	01 D6 00 08 B1 DD F9 D6 12 E7 4B DD 58 D9 7C DDK.X. .
1890	01 D6 00 08 B1 DD EA D9 35 D6 28 D6 98 DE B7 D75.(...
18A0	90 D8 88 E0 54 D8 A1 D9 F9 D6 83 D9 5F D8 09 D6	..T.....
18B0	F6 EA D9 5F D8 C1 D8 98 DE B7 D7 09 D6 F6 CF D8
18C0	01 D6 06 04 54 D8 54 D8 35 D6 28 D6 AC DE B7 D7T.T.5.(...
18D0	88 E0 65 E1 1B E1 9A EE FE D7 08 DF C0 D7 01 D6	..e.....
18E0	E8 03 10 D8 C1 D8 FE D7 30 E0 B7 D7 A0 D7 ED D60.....
18F0	09 01 D6 00 10 F9 E1 E3 D6 06 01 D6 00 10 E5 E1
1900	10 D8 35 D6 28 D6 FE D7 04 E0 C0 D7 F0 DE B7 D7	.5.(...
1910	00 E0 B7 D7 5F D8 2C E0 B7 D7 D8 DE B7 D7 C1 D8-
1920	35 D6 28 D6 F9 D6 31 D8 A1 D9 CC DC 09 D6 04 1E	5.(...1.....
1930	D7 F4 35 D6 28 D6 FE D7 88 D7 ED D6 09 09 D6 EC	..5.(...
1940	09 D6 FF E3 D6 05 09 D6 14 54 D9 CC DC 09 D6 28T....(
1950	57 DD 35 D6 28 D6 54 D9 54 D9 E8 DE 09 D6 14 05	W.5.(.T.T.....
1960	E1 22 EF 34 EF A0 DE C0 D7 54 D9 54 D9 EC DE 09	."4.....T.T....
1970	D6 14 05 E1 22 EF 34 EF A4 DE C0 D7 35 D6 28 D6"4.....5.(..

1980	F8	EC	20	ED	8C	DF	B7	D7	48	ED	7C	ED	00	E0	C0	D7H.
1990	35	D6	28	D6	90	DF	B7	D7	48	ED	F4	DE	B7	D7	FE	D7	5.(.....H.....)
19A0	F0	DE	B7	D7	7C	ED	5F	D8	FE	D7	88	D7	ED	D6	09	01
19B0	D6	00	40	54	D8	E3	D6	11	FE	D7	01	D6	00	40	A0	D7	..@T.....@..
19C0	ED	D6	06	01	D6	00	40	5F	D8	35	D6	28	D6	F9	D6	31@_5.(...1
19D0	D8	B7	D7	45	DD	09	D6	02	1E	D7	F4	35	D6	28	D6	54	...E.....5.(.T
19E0	D9	54	D9	E0	DE	09	D6	08	05	E1	CB	EF	10	DF	AD	D9	.T.....
19F0	54	D9	54	D9	E4	DE	09	D6	08	05	E1	CB	EF	14	DF	AD	T.T.....
1A00	D9	35	D6	28	D6	A8	E0	58	D9	44	DF	DF	D7	DD	EF	10	.5.(...X.D.....
1A10	DF	A1	D9	14	DF	A1	D9	3E	EE	7E	EF	2B	DA	92	EF	2B>.-.+....+
1A20	DA	04	EF	CA	EE	74	EE	E5	ED	54	EF	6C	E8	2B	DA	BBt...T.l.+..
1A30	EC	AF	E0	35	D6	37	F0	5F	5A	58	F7	F7	50	AD	97	FF	...5.7._ZX..P...
1A40	25	43	F0	59	58	05	01	00	83	D1	00	D1	F9	D1	D8	50	%C.YX.....P
1A50	AD	97	FF	25	28	D6	40	DE	B7	D7	76	D9	ED	D6	18	73	...%(.@...v....S
1A60	D8	FE	D7	40	DE	C0	D7	60	D7	ED	D6	0B	54	D9	3C	DE	...@...`....T.<.
1A70	DF	D7	09	D6	10	E5	E1	35	D6	28	D6	09	D6	04	54	D95.(....T.
1A80	F9	D6	A8	DE	31	D8	90	D8	54	D8	B7	D7	36	E9	90	D81...T...6...
1A90	FE	D7	4B	DD	DC	DE	31	D8	09	D6	04	CF	D8	54	D8	A1	..K...1.....T..
1AA0	D9	41	F0	FE	D7	E8	D7	54	D9	41	F0	45	DD	F3	D7	35	.A.....T.A.E...5
1AB0	F0	C0	DE	31	D8	90	D8	54	D8	C0	D7	08	D7	C4	35	D6	...1...T.....5.
1AC0	28	D6	C0	DE	FE	D7	A1	D9	10	D8	5F	D8	C4	DE	C0	D7	(.....-.....
1AD0	09	D6	04	54	D8	A1	D9	10	D8	5F	D8	C8	DE	C0	D7	35	...T.....-.....5
1AE0	D6	28	D6	A1	D9	B8	D9	54	D8	25	D8	25	D8	98	D7	ED	.(.....T.%.%....
1AF0	D6	02	11	D9	35	D6	28	D6	3C	DF	D3	D7	90	D8	54	D85.(.<....T.
1B00	C0	D7	35	D6	28	D6	FE	D7	E8	D7	98	DE	B7	D7	90	D8	..5.(.....
1B10	S4	D8	FE	D7	E8	D7	B7	D7	5F	D8	31	D8	A1	D9	5F	D8	T....._1....
1B20	F3	D7	F3	D7	B7	D7	54	D8	09	D6	04	54	D8	FE	D7	E8T....T....
1B30	D7	A1	D9	5F	D8	10	D8	C1	D8	F3	D7	B7	D7	54	D8	35_.....T.5
1B40	D6	28	D6	74	DF	B7	D7	65	E1	1B	E1	74	DF	B7	D7	04	.(.t...e...t....
1B50	F1	35	D6	28	D6	7C	DF	B7	D7	65	E1	1B	E1	7C	DF	B7	.5.(. ...e... ...
1B60	D7	04	F1	35	D6	28	D6	3C	DF	FE	D7	D3	D7	6A	D8	09	...5.(.<....j..
1B70	D6	03	3F	D7	10	D8	DF	D7	35	D6	28	D6	FE	D7	E8	D7	..?....5.(....
1B80	3C	DF	D3	D7	90	D8	54	D8	B7	D7	1A	D8	5F	D8	1B	D9	<.....T.....
1B90	48	E0	A0	D7	ED	D6	18	9A	D8	54	D9	54	D9	31	D8	09	H.....T.T.1..
1BA0	D6	08	05	E1	CB	EF	25	D8	45	DD	09	D6	05	57	DD	F3%E....W..
1BB0	D7	35	D6	28	D6	01	D6	00	02	09	D6	04	54	D9	F9	D6	.5.(.....T...
1BC0	A8	DE	31	D8	90	D8	54	D8	B7	D7	01	D6	E8	03	01	D6	..1...T.....
1BD0	80	3E	3F	D9	08	D7	E9	3F	D7	3F	D7	3F	D7	ED	D6	0B	.>?....??.?....
1BE0	E5	E1	54	D9	D0	DF	DF	D7	E3	D6	22	D0	DF	FE	D7	D3	..T.....".....
1BF0	D7	6A	D8	09	D6	12	29	D9	FE	D7	25	D8	DF	D7	09	D6	.j....)....%.....
1C00	12	70	D7	ED	D6	05	F9	E1	E3	D6	02	08	D8	35	D6	28	.P.....5.(
1C10	D6	FE	D7	1B	D9	3C	E0	B7	D7	A0	D7	ED	D6	4A	DC	DF<.....J..
1C20	D3	D7	ED	D6	24	FE	D7	1B	D9	40	E0	B7	D7	A0	D7	ED\$....@.....
1C30	D6	14	08	D8	C0	DE	7C	D8	E1	F0	53	F1	E4	DF	B7	D7S.....
1C40	01	D6	00	40	C1	D8	E3	D6	1C	FE	D7	C0	DE	7C	D8	E1	...@..... ...
1C50	F0	53	F1	01	D6	00	40	10	D8	C1	D8	E4	DF	C0	D7	58	.S....@.....X
1C60	D9	DC	DF	DF	D7	E3	D6	06	54	D9	DC	DF	DF	D7	35	D6T.....5.
1C70	28	D6	FE	D7	1B	D9	34	E0	B7	D7	A0	D7	ED	D6	46	D8	(.....4.....F.
1C80	DF	D3	D7	ED	D6	22	FE	D7	1B	D9	38	E0	B7	D7	A0	D7"....8.....
1C90	ED	D6	12	08	D8	C0	DE	E1	F0	41	F1	E0	DF	B7	D7	01A.....
1CA0	D6	00	40	C1	D8	E3	D6	1A	FE	D7	C0	DE	E1	F0	41	F1	..@.....A.

1CB0	01 D6 00 40 10 D8 C1 D8 E0 DF C0 D7 58 D9 D8 DF	...@.....X...
1CC0	DF D7 E3 D6 06 54 D9 D8 DF DF D7 35 D6 28 D6 B3T.....5.(..
1CD0	F1 79 F0 C0 F0 C4 DE B7 D7 41 F1 E0 DE 7A F1 F6	.Y.....A...z..
1CE0	F0 C8 DE B7 D7 53 F1 E4 DE 7A F1 F6 F0 65 F1 A8S...z...e..
1CF0	DE CC DE 09 D6 24 0F D9 58 D9 44 DE DF D7 35 D6\$..X.D...5.
1D00	28 D6 79 F0 C0 F0 C4 DE B7 D7 41 F1 E0 DE F6 F0	(.y.....A.....
1D10	C8 DE B7 D7 53 F1 E4 DE F6 F0 35 D6 28 D6 2B DAS.....5.(.+.
1D20	80 DE D3 D7 ED D6 F7 54 D9 80 DE DF D7 00 F3 80T.....
1D30	DE D3 D7 ED D6 37 54 D9 80 DE DF D7 CD F2 2B DA7T.....+.
1D40	34 DF D3 D7 FE D7 B4 DF DF D7 09 D6 03 3F D7 FE	4.....?..
1D50	D7 60 D7 ED D6 09 08 D8 03 F0 2B DA E3 D6 0C 09	..`.....+.....
1D60	D6 03 70 D7 ED D6 04 FE EA 2B DA 54 F0 2B DA E3	..p.....+..T.+..
1D70	D6 BD 35 D6 28 D6 11 DA B7 D7 D3 D7 58 D9 11 DA	..5.(.....X...
1D80	C9 D7 0E DA B7 D7 73 D8 0E DA C0 D7 74 E0 DF D7S.....t...
1D90	3E DA 35 D6 28 D6 3E DA 35 D6 9C F3 58 88 06 02	>.5.(.>.5...X...
1DA0	40 AD 97 FF 25 A7 F3 8A 06 03 40 25 02 00 34 02	@....%.....@%..4.
1DB0	50 AD 97 FF 25 28 D6 2B DA A5 F3 60 D7 ED D6 F7	P....%(.+....~....
1DC0	09 D6 25 9A F3 09 D6 03 54 D9 F9 D6 01 D6 FF 00	..%.....T.....
1DD0	D7 DA 31 D8 54 D8 DF D7 08 D7 F1 D7 DA 09 D6 03	..1.T.....
1DE0	86 DA 35 D6 28 D6 09 D6 24 9A F3 2B DA A5 F3 60	..5.(...\$..+....~
1DF0	D7 ED D6 F7 09 D6 64 9A F3 35 D6 28 D6 09 D6 40d..5.(...@
1E00	64 DE D3 D7 09 D6 10 70 D7 ED D6 0B E5 E1 54 D9	d.....p.....T.
1E10	70 DE DF D7 E3 D6 20 70 DE D3 D7 6A D8 09 D6 03	p..... p....j....
1E20	29 D9 FE D7 70 DE DF D7 09 D6 03 70 D7 ED D6 05)...p.....P....
1E30	F9 E1 E3 D6 02 E5 E1 54 D9 64 DE DF D7 35 D6 41T.d...5.A
1E40	F4 5F B8 00 00 BA 00 00 8A 15 52 B9 08 00 5A 52R...ZR
1E50	33 D0 83 E2 01 75 04 D1 E8 EB 07 D1 EA D1 D8 35	3....u.....5
1E60	01 20 5A D0 EA 52 E2 E6 5A 47 5A 4A 52 75 D9 5A	. Z..R..ZGZJRu.Z
1E70	50 AD 97 FF 25 28 D6 B5 F3 50 DE 6A D8 5F D8 FE	P...%(...P.j._..
1E80	D7 86 D8 74 DE DF D7 09 D6 03 54 D9 F9 D6 50 DE	...t.....T...P.
1E90	6A D8 1A D8 86 DA 08 D7 F5 08 D8 FB F3 E4 F3 35	j.....5
1EA0	D6 28 D6 1A D8 DF D7 6A D8 35 D6 28 D6 1A D8 C0	.(.....j.5.(....
1EB0	D7 7C D8 35 D6 28 D6 B5 F3 24 E0 09 D6 07 86 DA	. ..5.(...\$.....
1EC0	50 DE 6A D8 58 D9 A1 F4 09 D6 02 A1 F4 09 D6 03	P.j.X.....
1ED0	A1 F4 A8 DE 09 D6 0C 05 E1 F9 D6 31 D8 B7 D7 AB1....
1EE0	F4 09 D6 02 1E D7 F4 D8 DE B7 D7 AB F4 18 DF B7
1EF0	D7 AB F4 30 DE B7 D7 AB F4 1C DF B7 D7 AB F4 C4	...0.....
1FO0	DF B7 D7 09 D6 02 D8 E1 AB F4 C0 DE 09 D6 08 05
1F10	E1 F9 D6 31 D8 B7 D7 AB F4 09 D6 02 1E D7 F4 09	..1.....
1F20	D6 0D A1 F4 09 D6 0A A1 F4 50 DE 6A D8 5F D8 50P.j._.P
1F30	DE 6A D8 10 D8 86 DA E4 F3 35 D6 28 D6 05 D6 03	.j.....5.(....
1F40	54 D9 F9 D6 54 DE 31 D8 90 D8 54 D8 B7 D7 01 D6	T...T.1...T.....
1F50	00 01 A4 D8 10 D8 25 D8 09 D6 02 54 D9 F9 D6 1A%....T....
1F60	D8 A1 F4 10 D8 09 D6 10 70 D7 ED D6 05 09 D6 10p.....
1F70	A1 F4 08 D7 EA 2B DA 08 D7 CA 35 D6 28 D6 50 DE+....5.(.P.
1F80	6A D8 09 D6 10 A1 F4 09 D6 02 A1 F4 10 D8 A1 F4	j.....
1F90	60 DE D3 D7 A1 F4 3B F5 09 D6 10 A1 F4 09 D6 03	~.....;.....
1FA0	A1 F4 FE D7 50 DE 09 D6 03 54 D8 5F D8 50 DE 09P....T._.P..
1FB0	D6 03 54 D8 3F F4 1A D8 C0 D7 7C D8 01 D6 FF 00	..T.?.....
1FC0	A1 F4 75 F4 35 D6 28 D6 7C D8 10 D8 6A D8 D3 D7	..u.5.(. ...j...
1FD0	60 DE DF D7 A0 DE B7 D7 54 DE C0 D7 A4 DE B7 D7	~.....T.....

1FE0	54 DE 7C D8 C0 D7 30 DE B7 D7 54 DE 09 D6 04 54	T. ...0...T....T
1FF0	D8 C0 D7 54 D9 30 DE 7C D8 C0 D7 35 D6 28 D6 7C	...T.0. ...5.(.
2000	D8 10 D8 6A D8 D3 D7 60 DE D3 D7 70 D7 27 D9 ED	...j...`...p.'..
2010	D6 0C 86 D8 58 DE B7 D7 6A D8 10 D8 C6 F5 35 D6X...j....5.
2020	28 D6 58 DE B7 D7 6A D8 FE D7 D3 D7 FE D7 09 D6	(.X...j.....
2030	03 3F D7 09 D6 02 70 D7 ED D6 05 FD F5 E3 D6 02	.?....p.....
2040	C6 F5 7C F5 35 D6 28 D6 B5 F3 24 E0 09 D6 40 865.(...\$...@.
2050	DA E4 F3 35 D6 28 D6 58 DE B7 D7 6A D8 D3 D7 FE5.(.X...j....
2060	D7 09 D6 30 3F D7 09 D6 10 DA D8 34 DE D3 D7 09	...0?....4....
2070	D6 03 3F D7 70 D7 ED D6 40 FE D7 09 D6 07 3F D7	..?..p...@.....?
2080	FE D7 58 D9 70 D7 10 D8 09 D6 02 70 D7 4A D7 27	..X.p.....p.J.'
2090	D9 ED D6 22 09 D6 0F 3F D7 FE D7 09 D6 07 70 D7"....?.....p.
20A0	ED D6 07 08 D8 B5 F4 E3 D6 0A 09 D6 06 70 D7 EDp..
20B0	D6 02 46 F6 54 D9 E3 D6 02 27 D9 35 D6 28 D6 5C	..F.T....'5.(.\
20C0	DE B7 D7 58 DE B7 D7 B8 D9 5F D8 10 D8 6A D8 3F	...X....._...j.?.
20D0	F4 2B DA 10 D8 6A D8 B7 D7 70 D7 35 D6 28 D6 2B	.+...j...p.5.(.+
20E0	DA 54 D9 5C DE C0 D7 50 DE 09 D6 10 54 D8 10 D8	.T.\...P....T...
20F0	F9 D6 31 D8 D3 D7 09 D6 10 70 D7 ED D6 23 31 D8	.1.....p...#1.
2100	6A D8 D3 D7 09 D6 03 70 D7 ED D6 0F 31 D8 6A D8	j.....p....1.j.
2110	5C DE C0 D7 58 D9 95 D9 E3 D6 03 09 D6 02 E3 D6	\...X.....
2120	02 58 D9 1E D7 CC 5C DE B7 D7 2B DA 35 D6 28 D6	.X....\....+5.(.
2130	2B DA 54 D9 58 DE C0 D7 50 DE 09 D6 08 54 D8 10	+.T.X...P....T..
2140	D8 F9 D6 31 D8 D3 D7 09 D6 10 70 D7 ED D6 23 31	.1.....p...#1
2150	D8 6A D8 D3 D7 09 D6 02 70 D7 ED D6 0F 31 D8 6A	.j.....p....1.j
2160	D8 58 DE C0 D7 58 D9 95 D9 E3 D6 03 09 D6 02 E3	.X...X.....
2170	D6 02 58 D9 1E D7 CC 58 DE B7 D7 2B DA 35 D6 28	..X....X...+5.(
2180	D6 50 DE 6A D8 2E F7 FE D7 ED D6 0E 6A D8 DD F6	.P.j.....j...
2190	ED D6 05 BD F6 E3 D6 02 54 D9 35 D6 28 D6 7F F7T.5.(....
21A0	ED D6 07 55 F6 ED D6 02 20 F6 35 D6 28 D6 50 DE	...U....5.(.P.
21B0	09 D6 28 67 D9 50 DE 6A D8 09 D6 27 9C DA 50 DE	..(g.P.j...'.P.
21C0	D3 D7 ED D6 02 9C F7 2B DA E3 D6 E2 35 D6 5B 59+....5.[Y
21D0	58 CF D4 F7 B8 00 00 CF FF 06 A4 01 CF 50 51 53	X.....PQS
21E0	BB FF 06 8A 06 00 40 F6 47 14 80 79 1C 87 7F 16@.G..y....
21F0	88 05 47 87 7F 16 FF 47 18 FF 47 14 75 09 C6 06	..G....G..G.u...
2200	AD 00 00 C7 07 FF D7 EB 10 F6 06 E2 00 01 74 09t.....t.
2210	88 06 E0 00 C6 06 E2 00 00 EB B3 8B 47 14 0B C0G...
2220	7E 24 87 77 16 AC 87 77 16 88 06 00 40 FF 4F 14	-\$..w...w....@.O.
2230	75 06 C7 07 FF D7 EB 0E 8A 47 14 38 06 E4 00 75	u.....G.8...u
2240	05 C6 06 E2 00 01 C3 50 51 53 BB FF 06 8A 06 01PQS.....
2250	40 8A C8 A8 20 74 03 E8 C1 FF F6 C1 10 74 23 8A	@....t.....t#.
2260	06 03 40 24 02 34 02 75 15 F6 06 E1 00 02 74 0E	..@S.4.u.....t.
2270	C6 06 AD 00 01 C7 47 14 00 00 C7 07 FF D7 88 06G.....
2280	E1 00 E9 49 FF 50 51 53 9C 8A 06 AD 01 40 25 07	...I.PQS.....@%.
2290	00 88 06 AD 01 C6 06 AC 01 01 C6 06 AE 01 01 C7
22A0	06 A6 01 00 00 9D E9 25 FF 28 D6 E4 EB 09 D6 08%.(....
22B0	4B DD B8 DF AD D9 4B E2 09 D6 08 4B DD C0 DF AD	K.....K....K....
22C0	D9 35 D6 28 D6 20 DF D3 D7 09 D6 0A 98 D7 ED D6	.5.(.
22D0	36 54 D9 B4 DE C0 D7 54 D9 B8 DE C0 D7 09 D6 04	6T.....T.....
22E0	54 D9 F9 D6 09 D6 04 F8 E4 B4 DE C9 D7 09 D6 06	T.....
22F0	F8 E4 B8 DE C9 D7 08 D7 EB B4 DE B7 D7 B8 DE B7
2300	D7 0C E0 AD D9 92 E9 35 D6 28 D6 09 D6 28 81 E15.(....(.

2310	35 D6 28 D6 54 D9 42 EA 01 D6 10 27 54 D8 58 D9	5. (. T.B....'T.X.
2320	42 EA 08 D8 BC DE A1 D9 35 D6 28 D6 AC DB B7 D7	B.....5. (.....
2330	FE D7 01 D6 A8 E4 98 D7 ED D6 04 08 D8 54 D9 35T.S
2340	D6 28 D6 20 DF D3 D7 09 D6 0A 98 D7 ED D6 41 54	.(.....AT
2350	D9 B4 DE C0 D7 54 D9 B8 DE C0 D7 54 D9 00 DF C0T.....T...
2360	D7 54 D9 04 DF C0 D7 09 D6 04 54 D9 F9 D6 92 E5	.T.....T....
2370	08 D7 FB 00 DF B7 D7 04 DF B7 D7 0C E0 AD D9 FE
2380	EA 42 E4 01 D6 10 27 54 D9 F9 D6 4B E2 08 D7 FB	.B....'T...K....
2390	35 D6 28 D6 F9 D6 31 D8 31 D8 B8 D9 1C E4 42 E4	5. (..1.1....B.
23A0	09 D6 78 54 D9 F9 D6 4B E2 08 D7 FB 08 D7 E7 35	..xT...K.....5
23B0	D6 28 D6 C3 F8 2A F9 4B E2 12 F9 FE D7 FE D7 B8	.(...*K....
23C0	D9 1C E4 42 E4 09 F9 92 F9 4B E2 41 F9 A9 F8 35	...B....K.A...5
23D0	D6 A1 D6 FF 06 2E EA FF 05 00 00 00 00 00 7E 06 0A~..
23E0	00 76 F3 96 F3 A1 D6 FF 05 2E EA FF 04 00 00 00	.v.....
23F0	00 BE 05 0A 00 A1 D6 FF 04 2E EA FF 03 00 00 00
2400	00 BE 04 0A 00 A1 D6 FF 03 2E EA FF 06 00 00 00
2410	00 BE 03 0A 00 A1 D6 D8 F7 00 00 DD F7 00 00 47G
2420	F8 00 00 DD F7 00 00 28 D6 FE D7 B7 D7 54 D8 09(.....T..
2430	D6 04 54 D8 FE D7 60 DF C0 D7 FE D7 B7 D7 54 D8	..T...`.....T.
2440	09 D6 04 54 D8 FE D7 64 DF C0 D7 FE D7 B7 D7 54	...T...d.....T
2450	D8 09 D6 04 54 D8 FE D7 68 DF C0 D7 FE D7 B7 D7T...h.....
2460	54 D8 09 D6 04 54 D8 FE D7 6C DF C0 D7 64 DF B7	T....T....l...d..
2470	D7 B7 D7 68 DF B7 D7 B7 D7 CF D8 9A D8 7C D8 54	...h..... ..T
2480	D8 FE D7 70 DF C0 D7 FE D7 B7 D7 54 D8 09 D6 04	...p.....T....
2490	54 D8 FE D7 74 DF C0 D7 FE D7 B7 D7 54 D8 09 D6	T....t.....T...
24A0	04 54 D8 FE D7 78 DF C0 D7 FE D7 B7 D7 54 D8 09	.T...x.....T..
24B0	D6 04 54 D8 FE D7 7C DF C0 D7 FE D7 B7 D7 54 D8	..T...T.
24C0	09 D6 04 54 D8 FE D7 80 DF C0 D7 FE D7 B7 D7 54	...T.....T
24D0	D8 09 D6 04 54 D8 FE D7 84 DF C0 D7 FE D7 B7 D7T.....
24E0	54 D8 09 D6 04 54 D8 FE D7 88 DF C0 D7 FE D7 B7	T....T.....
24F0	D7 54 D8 09 D6 04 54 D8 FE D7 8C DF C0 D7 84 DF	.T....T.....
2500	B7 D7 B7 D7 88 DF B7 D7 B7 D7 CF D8 9A D8 7C D8
2510	54 D8 90 DF C0 D7 35 D6 28 D6 24 E0 09 D6 70 54	T....5.(\$...pT
2520	D8 FE D7 54 DF C0 D7 09 D6 6A 54 D8 FE D7 58 DF	...T....jt...X.
2530	C0 D7 09 D6 16 54 D8 FE D7 5C DF C0 D7 27 FA 35T...`....'5
2540	D6 28 D6 28 E0 A1 D9 01 D6 00 01 45 DD 01 D6 00	.(.(...E....
2550	02 57 DD 35 D6 57 FB C6 06 03 60 34 C6 06 03 60	.W.5.W....`4...
2560	56 C6 06 03 60 B0 58 88 06 00 60 88 26 00 60 C6	V...`X...`.&`.
2570	06 02 60 00 C6 06 02 60 00 C6 06 01 60 50 AD 97`....`P..
2580	FF 25 28 D6 09 D6 3C 78 E0 DF D7 01 D6 A2 00 80	.%(...<x.....
2590	E0 DF D7 09 D6 64 7C E0 DF D7 74 E0 D3 D7 08 D8dt.....
25A0	74 E0 D3 D7 08 D8 78 E0 D3 D7 08 D8 35 D6 B0 FB	t.....x.....5...
25B0	FB AD 97 FF 25 B7 FB 80 0E EB 00 40 8A 06 EB 00%.....@....
25C0	88 06 00 80 AD 97 FF 25 28 D6 09 D6 10 3C DE D3%(...<..
25D0	D7 6A D8 01 D6 FF 00 29 D9 FE D7 3C DE DF D7 09	.j.....)<....
25E0	D6 05 98 D7 ED D6 05 E5 E1 E3 D6 02 F9 E1 01 D6
25F0	60 09 40 DE C0 D7 35 D6 FA FB 8A 06 02 80 24 3F	`.@...5.....\$?
2600	88 06 A4 00 AD 97 FF 25 28 D6 82 FB 41 FB 55 FB%(...A.U.
2610	4B E2 35 D6 28 D6 2C DE 01 D6 2C 01 B8 D9 67 D9	K.5.(.,...,g.
2620	4B E2 54 D8 01 D6 2C 01 67 D9 4B E2 35 D6 28 D6	K.T...,g.K.5.(.
2630	01 D6 10 27 D0 DE C0 D7 01 D6 10 27 D8 DE C0 D7	...`.....`....

2970	FF
2980	FF
2990	FF
29A0	FF
29B0	FF
29C0	FF
29D0	FF
29E0	FF
29F0	EA AF FD 00 00 FF
2A00	-----

```
:101600000203D6AD50AD97FF250BD6AC9850AD97E1
:10161000FF25FFFFFFF9A
:10162000FFFFFFFFFFF47474D4D8976008B10
:10163000F7AD97FF2537D68B76004545AD97FF254B
:10164000FFFFFFF9A
:10165000FFFFFFF9A
:10166000FFFFFFF8A
:10167000FFFFFFF7A
:10168000FFFFFFF6A
:10169000FFFFFFF5A
:1016A000FF474757AD97FF254D4D8976005E474769
:1016B00057AD97FF25FFFFFFF76
:1016C000FFFFFFF7502AD97FF25FFFFFFF45
:1016D000FFFFFFF8A45029803C350ADE6
:1016E00097FF25E5D6AC9803F0AD97FF25EFD658C8
:1016F0000BC074F146AD97FF25FBD683ED048F46F2
:1017000008F4602AD97FF250AD78B4600403B4627
:1017100002731C894600AC9803F0AD97FF2520D7D3
:10172000588BD00346008BF82B7E0233D778E483A6
:10173000C50446AD97FF2539D758034600EBCF4186
:10174000D75A5823C250AD97FF254CD75A580BC2D1
:1017500050AD97FF2557D75A5833C250AD97FF2544
:1017600062D7580BC0B8000075014050AD97FF25F7
:1017700072D75A583BC2EBED7AD7580BC0B800006D
:101780007E014050AD97FF258AD7580BC0B80000A6
:1017900079014050AD97FF259AD75A582BC2EBEDEF
:1017A000A2D7585A2BC2EBE5AAD75A582BC92BC23D
:1017B00013C951AD97FF25B9D75FFF35AD97FF2509
:1017C000C2D75F58ABAD97FF25CBD75F580105ADAA
:1017D00097FF25D5D75F2BC08A0550AD97FF25E130
:1017E000D75F58AAAD97FF25EAD74D4D8F4600AD7C
:1017F00097FF25F5D7FF76004545AD97FF2500D823
:101800008BFCFF35AD97FF250AD84444AD97FF25E3
:1018100012D85A585250AD97FF251CD88BFCFF7533
:1018200002AD97FF2527D85F5A58525750AD97FF02
:101830002533D8FF7600AD97FF253CD8588BFC8B1D
:101840000D2BC175070501004444EB03B80000509F
:10185000AD97FF2556D85A5803C250AD97FF256162
:10186000D85A582BC250AD97FF256CD8584050AD70
:1018700097FF2575D8584850AD97FF257ED858401A
:101880004050AD97FF2588D858484850AD97FF2560
:1018900092D858D1E050AD97FF259CD858D1F85038
:1018A000AD97FF25A6D85F582BD2F7F75250AD97FFCA
:1018B000FF25B4D85F5A58F7E2F7F75250AD97FFBB
:1018C00025C3D85F5A58F7EAF7FF50AD97FF25D1E7
:1018D000D85F58F7E750AD97FF25DCD85F5899F7E8
:1018E000FF50AD97FF25E8D85F2BD258F7F752ADE0
:1018F00097FF25F5D8595F58E31196D1CFD1C7731B
:1019000002A449D1E9F3A57301A496AD97FF25F58B
:10191000D813D958F7D850AD97FF251DD9580BC00B
:1019200078F250AD97FF2562D72BD9585A3BC27930
:10193000019252AD97FF2539D9585A3BD0EBF0416F
:10194000D95A59582BFF3BC279053BC178014757FB
:10195000AD97FF25C3D60000C3D601005ED9585904
:101960005FF3AAAD97FF2569D92BC0EBF26FD98B36
:10197000C450AD97FF2578D9580BC074015050ADBS
:1019800097FF2585D9FF7602AD97FF258ED9FF7683
:1019900004AD97FF2597D98B4600894602AD97FF86
:1019A00025A3D95FFF7502FF35AD97FF25AFD95F3E
:1019B00058AB58ABAD97FF25BAD98BFCFF7502FF2A
:1019C00035AD97FF25C7D983C404AD97FF25D0D97E
```

```
:1019D0005A5F5B5957525150AD97FF25DED98BFCAD
:1019E000FF7506FF7504AD97FF25ECD9FF7602FF62
:1019F000760083C504AD97FF25D8D600D8D608D881
:101A0000D60AD8D60CD8D60ED8D610D8D612D8D654
:101A100014D8D616D8D618C3D6FFD75B4B4BC707FA
:101A20002EEA8B67065E5DAD97FF252FDA4E4EC717
:101A30007FFD75556896706BF1BDAFF6F0233DAF7
:101A400028D6F9D95FD810D8B7D754D835D650DAB8
:101A5000935F8B1DC707FFD7C7471400008B7F0814
:101A600083EF068935895D02C745040000897F063A
:101A700093E9C3FB76DA5F4F4FFF257DDA5F8B3D3D
:101A80008BCFE314EBF188DA588F47160BC07E0832
:101A90008947148B7F0CEBDFA97FF259EDAC74794
:101AA00018000058F7D88947148F47168B7F0EEB24
:101AB000C628D608DA7BDA35D628D60BDA7BDA35B3
:101AC000D628D66DD958D986DA08D835D6CFDAFFD8
:101AD000B798EFAD97FF25D9DA8B4708051000506E
:101AE000AD97FF25E6DA5F2BC08A05475750AD97C3
:101AF000FF25F4DAB020E966FE20202020202020F7
:101B00002003DB2BFF580BC07810BF08002BC750F9
:101B1000BAF9DA52790203F8574757AD97FF2528EB
:101B2000D601DBEDD60586DAE3D6F635D628D658CB
:101B3000D91FD835D637DBFF4F168B7F1658AAAD82
:101B400097FF2545DB583C0A72020407043050AD6C
:101B500097FF2528D6D7DA11DAC0D735D628D6C5CB
:101B6000D911DAB7D7D7DA1AD85FD835D628D6251B
:101B7000D888D7EDD60509D62D35DB35D628D61031
:101B8000D8FFD9B7D7A4D810D843DB35DB10D83568
:101B9000D628D67DBB1AD860D7EDD6F735D628D62D
:101BA000FED718D954D953DB91DB6DDB5DDB35D61A
:101BB00028D69EDB86DA2DDB35D628D6B7D7B0DB24
:101BC00035D628D610D854D953DB91DB5DDB25D828
:101BD0001AD85FD81FDB86DA35D628D654D9C2DBAF
:101BE0002DDB35D6E6DB8B7E008A0598400146006A
:101BF000580BC07503E93FFA57AD97FF25FFDB8B04
:101C000067082BC050BE0000AD97FF2528D6E4DB47
:101C1000CDDA7CD8E4DA86DA2DDBE4DA86DAB1DAFA
:101C2000FDDB28D658D9E4DBE4DA86DA35D630DCB9
:101C3000965F595ED1E9F3A77407B901007302F703
:101C4000D95196AD97FF2549DC96595F5E03F94F50
:101C500003F14EFD3A4FC96AD97FF255EDC585FC3
:101C60005703F84F803D2075034875F750AD97FF37
:101C70002573DC2BD25F47578A053C3A72063C41FC
:101C8000720E2C072C3072083A470A7303985042A0
:101C900052AD97FF2597DC5F5A585952F7670A9162
:101CA000F7670A03C713CA5A505152AD97FF252848
:101CB000D671DCEDD60B95DC58D911DAC9D7E3D64D
:101CC000F035D6C5DC8B3E6205E95EF9CEDC5A58AC
:101CD0008BFC0145021115AD97FF25DDDC5A58F745
:101CE000D883D200F7DA5052AD97FF2528D6DBDC37
:101CF000CCDC35D628D660D710D860D73FD735D6BC
:101D000028D6ECD8D710D808D835D628D6FED708
:101D100088D7EDD602DBDC35D628D6ECD8F4DC3512
:101D2000D628D6DCD9ECD8FED788D7EDD605CCDCBE
:101D3000E3D602C5D935D628D6DCD9DCD921DDECED
:101D4000DCCCDC35D647DD5899EB854DDD585AF7AC
:101D5000EA5052AD97FF2559DD5F5A58F7FF50AD55
:101D600097FF2565DD585AF7E25052AD97FF257170
:101D7000DD5F5A58F7F75250AD97FF257EDD897623
:101D8000F895EFC5B5E5F8BC7F7E3508BCA8BC638
:101D9000F7E303C883D2000BF679022BD30BDB7970
:101DA000042BCF1BD651528B5EFC8B76FEAD97FF7A
```

```
:101DB00025B3DD5F5A58590BFF9C7902F7DF0BD230
:101DC000790203D7F7F791F7F79D7907F7D883D111
:101DD00000F7D95051AD97FF2528D6E8D77CDDF321
:101DE000D7B1DD35D6E7DD58592BD2F7770A91F711
:101DF000770A505152AD97FF2528D6E5DD43DB35F4
:101E0000DB35D628D6F9DB8D9F4DCEDD6F735D6F2
:101E100028D610D81AD80CDD53DB03DE6DBB5DBB72
:101E200035D628D610DE86DA2DBB35D6C3D6A0000F
:101E3000C3D6A000C3D6A400C3D6A500C3D6A700AE
:101E4000C3D6A800C3D6AA00C3D6AB00C3D6AC0085
:101E5000C3D6AD00C3D6D500C3D6DB00C3D6DD00E4
:101E6000C3D6DF00C3D6E000C3D6E100C3D6E2008C
:101E7000C3D6E300C3D6E400C3D6E500C3D6E6006C
:101E8000C3D6E700C3D6E800C3D6E900C3D6EA004C
:101E9000C3D6EB00C3D6EC00C3D6EE00C3D6F00029
:101EA000C3D6F200C3D6F400C3D6F600C3D6FE00F4
:101EB000C3D60001C3D60201C3D60401C3D60601AE
:101EC000C3D60E01C3D61601C3D61801C3D61A0154
:101ED000C3D63E01C3D64001C3D64201C3D6440196
:101EE000C3D65401C3D65C01C3D66401C3D67801FE
:101EF000C3D68C01C3D68E01C3D69001C3D692013E
:101F0000C3D69401C3D69601C3D69801C3D69A010D
:101F1000C3D69C01C3D6A001C3D6A401C3D6A601D3
:101F2000C3D6A801C3D6A901C3D6AA01C3D6AB01A3
:101F3000C3D6AC01C3D6AD01C3D6AE01C3D6AF0183
:101F4000C3D6B001C3D6B101C3D6B201C3D6B30163
:101F5000C3D6B401C3D6B501C3D6B701C3D6B90140
:101F6000C3D6BB01C3D6BD01C3D6BF01C3D6C10111
:101F7000C3D6C301C3D6C501C3D6C701C3D6C901E1
:101F8000C3D6CB01C3D6CD01C3D6CF01C3D6D101B1
:101F9000C3D6D301C3D6D501C3D6D701C3D6D90181
:101FA000C3D6DD01C3D6E101C3D6E501C3D6E9013D
:101FB000C3D6ED01C3D6F101C3D6F201C3D6F601F3
:101FC000C3D6F801C3D6FC01C3D6FE01C3D6FF01B8
:101FD000C3D60002C3D60102C3D60202C3D603028F
:101FE000C3D60402C3D60602C3D60802C3D60A0269
:101FF000C3D60C02C3D61002C3D61402C3D618022D
:10200000C3D61C02C3D61E02C3D62002C3D62402E6
:10201000C3D62802C3D62C02C3D63002C3D634029C
:10202000C3D63602C3D600C0C3D660C0C3D664C010
:10203000C3D666C0C3D668C0C3D66AC0C3D66CC098
:10204000C3D66EC0C3D60008C3D60014C3D6008062
:10205000C3D60180C3D60280C3D60060C3D6016058
:10206000C3D60260C3D60360C3D60000C3D6010046
:10207000C3D64000C3D60040C3D60140C3D60240F9
:10208000C3D60340C3D65000A1D60E00C0EB29F53D
:10209000D5FED007390F0B15A6199CE090AD97FF20
:1020A00025A3E090AD97FF25AAE090AD97FF25B15D
:1020B000E090AD97FF25B8E090AD97FF25BFE09089
:1020C000AD97FF25C6E090AD97FF25CDE058595A52
:1020D000525150AD97FF25D9E058595A525150AD41
:1020E00097FF25E5E058595A525150AD97FF2528E2
:1020F000D609D610FFD9C0D735D628D609D60AFFBB
:10210000D9C0D735D628D61AD854D810D835D6281D
:10211000D6B7D790D854D8B7D735D628D6E8D794DD
:10212000DEB7D71AD831D894DE0FE198D727D9ED8A
:10213000D60598DEE3D6029CDEC0D798DEB7D79CE2
:10214000DEB7D71AD85FD8FED7E8D79AD854D89434
:10215000DEC0D7F3D709D60298D72BDAEDD6C0F375
:10216000D708D835D628D6FED7B7D79AD8FED79C69
:10217000DEC0D79AD894DEC0D758D998DEC0D735FC
:10218000D683E1580806EB008A06EB00880600803B
```

FMQ-13 Object Code, "Digital Wind Sensor Application Program" CPIN 83M-FMQ13-F002-00A 31 May 1990

```
:10219000AD97FF2596E15B34FF2006EB008A06EB49
:1021A0000088060080AD97FF25ABE15F5A58F7F72E
:1021B00050AD97FF25B7E1595805010083D100D1F3
:1021C000F9D1D850AD97FF25CAE159585AD1F8D165
:1021D000DAE2FA52AD97FF25DAE15958D1E8E2FC8C
:1021E00050AD97FF25E7E159BFA000854D02750569
:1021F00083F1FF210DAD97FF25FB159BFA00085BD
:102200000D7505090D094D02AD97FF2528D654D946
:102210001CDFB7D701D6C409A0D7EDD60527D9E36F
:10222000D60658D91CDFC9D735D62CE2B855558B00
:10223000C8BAAAAAA03C2790433C1D1E82BC875046D
:10224000D1E22BC20BC150AD97FF254DE28A06B2F9
:1022500001E650AD97FF2528D64CDFFD7D3D76ACD
:10226000D809D60229D9FED725D8DFD735D628D622
:1022700058D92AE258D970D7EDD605E5E1E3D60260
:10228000F9E14BE22BDA35D68AE2BA0000B9080050
:102290005FFA8A05C60555803D55750BC605AA80AF
:1022A0003DAA740142EB0142880547FBE2E352ADCF
:1022B00097FF2528D654D938DFDFD709D60454D95B
:1022C00001D6000854D9F9D62BDA4BE238DFD3D740
:1022D0000CE24AD7EDD6F154D938DFDFD731D888B0
:1022E000E2FED7EDD60295D954D84BE22BDA09D6C7
:1022F000081ED7D4EDD605F9E1E3D602E5E14BE2BD
:102300002BDA35D606E38BFE5E5A2BC0B90800AC3B
:1023100003D0E2FB8BF752AD97FF251DE32BC08A5C
:10232000060280248050AD97FF2528D609D6025496
:10233000D909D6F801D600C0F9D631D804E34BE26A
:102340002BDA09D6081ED7F209D6FEB7D770D7ED1B
:10235000D607E5E1A1E0E3D602F9E14BE22BDA355D
:10236000D663E3FAC6060360008A0E00608A0600A0
:10237000608AE851B93000E2FEC6060360008A0EAA
:102380000608A0600608AE8FB51AD97FF2528D6D9
:1023900054D938DFDFD72BDA4BE238DFD3D70CE262
:1023A0004AD7EDD6F109D62061E3B8D9A8D727D905
:1023B000EDD61D5FD809D60209D6063FD90CE22713
:1023C000D93FD7EDD605E5E1E3D602F9E1E3D6043E
:1023D000C5D908D84BE22BDA35D628D601D600402D
:1023E0001BE3EDD60BE5E154D94CDFDFD7E3D61183
:1023F00057E209D60270D7EDD605F9E1E3D6020817
:10240000D84BE22BDA35D628D64BE22BDA2AE38EEC
:10241000E3DAE3B3E26EE2E3D6EF35D61EE4C606B6
:10242000E50000B9040058BA01000216E500EE8A82
:10243000C442EE8006E50010E2EC8806B201AD97DA
:10244000FF2544E48A06B201E640C606B10100ADAC
:1024500097FF2555E48026EB00F88A06EB008806F6
:102460000080C606E80001C606E90000B90400BF06
:10247000F600C70500004747E2F8C6060280008A5A
:1024800006EB000A06E80088060080C606A9010FD0
:102490009BC606A90100C606A801002BC08A06003B
:1024A000808AE08A060180C606028000B90400D353
:1024B000E88A0EEA00D1E18BF90185F600FE06EA12
:1024C000008026EA0003F606E80003750FFE06E921
:1024D00000F606E900037504AD97FF25FE06E80047
:1024E0008026E800038A06EB000A06E8008806005A
:1024F00080C606A9010FEB98FAE48A06EB00590A98
:10250000C188060080B90500E2FE880602802BC063
:10251000C606A9010F9BC606A90100C606A80100B0
:102520008A0600808AE08A060180B90400D3E85058
:10253000AD97FF2536E58A06B4010AC0753EC6068A
:102540000360E48A060160A880752FC6060360D880
:102550008B069A01890600018A0602608A260260BB
:10256000F7D889069A01C6060360B02BC088060218
```

```
:102570006088060260C606B40101EB12C60603605D
:10258000E48A060160A8807405C606B40100AD9710
:10259000FF2528D609D604F8E4B4DEC9D709D60544
:1025A000F8E404DFC9D709D606F8E4FED7B8DEC9D7
:1025B000D7F8DEC0D709D607F8E400DFC9D735D68B
:1025C00028D601D60001F8DEB7D701D6990501D685
:1025D000CD0C3FD9EDD605E5E1E3D602F9E135D6DC
:1025E00028D62BDA30DFD3D7EDD61B54D930DFDF36
:1025F000D753E492E544DFD3D7EDD60242E4C0E5F9
:1026000058D980DDED7E3D6D935D60DE683EC087E
:102610008BFC8B450E8B4D0AF7E1895504894506E5
:1026200031C089450289058B450CF7E1014504114C
:1026300055028315008B4D088B450EF7E1014504CB
:102640001155028315008B450CF7E1014502111568
:10265000FA83C410FF7506FF7504FF7502FF35FB92
:10266000AD97FF2566E68BFC8B450601450E8B4535
:102670000411450C8B450211450A8B051145088351
:10268000C408AD97FF25C3D600408CE65859F7E93A
:10269000D1E0D1D2D1E0D1D252AD97FF259FE659FA
:1026A0005A2BC0D1FAD1D8D1FAD1D8F7F950AD9779
:1026B000FF25B4E65F582BD2F7F750AD97FF25C141
:1026C000E658D1E0D1E07902F7D82D0040F7D85094
:1026D000AD97FF25D6E6582D001050AD97FF252861
:1026E000D6BFE6FED7FED78AE6FED709D6B98AE678
:1026F00001D6150554D81AD88AE601D6AAD654D8D8
:102700008AE601D6882454D81AD88AE654D82BDA17
:1027100035D628D6D4E6DFE635D61CE758D1C872C0
:1027200001ADD15600AD97FF2500F4000400340040
:10273000C4001400E400D4002428D6FED788D7E8CB
:10274000D71BD910D8FED788D71AE711D9B8D9A086
:10275000D71AE710D89DE6FED701D690E554D810D9
:10276000D801D6821A8AE686E654D89DE6FED7FEC0
:10277000D78AE6FED701D6B6018AE601D6A0FC5478
:10278000D88AE601D62F0A54D88AE6F3D790D80122
:10279000D629E754D8B7D754D81BD92BDA35D62841
:1027A000D601D668019DE635D628D654D901D65033
:1027B0004601D60020D9DD08D835D628D601D60066
:1027C000201AD898D7EDD60486E65FD835D6D0E75C
:1027D0005A58562BFF8BF7B91000D1E0D1D2D1D681
:1027E000D1E0D1D2D1D6D1E7D1E7473BF772032B65
:1027F000F747D1EFE2E45E57AD97FF25FEE783ECA4
:10280000048BFC31D231C08905894502B92000D141
:10281000650AD15508D15506D15504D1D0D1D2D1B0
:10282000650AD15508D15506D15504D1D0D1D2D1A0
:1028300025D15502D125D1550283050183550200CA
:102840003B5502721275043B05720C2B051B550299
:1028500083050183550200D16D02D11DE2B18B05C4
:102860008B550283C40C5052AD97FF2528D634DE19
:10287000D3D709D6203FD7EDD60601D6800094E104
:1028800035D628D601D68000ACDEB7D7C8DFD3D77F
:10289000EDD61D01D668F701D6DC053FD9EDD6058A
:1028A00081E1E3D60894E154D9C8DFDFD7E3D61A33
:1028B00001D630F801D6E8033FD9EDD60B81E158B7
:1028C000D9C8DFDFD7E3D60208D835D628D65FD8F7
:1028D0004BDD01D6002054D9CCDC09D60EC8E13539
:1028E000D628D64BDD01D6004054D9CCDC09D60D14
:1028F000C8E154D809D602DBE135D628D698DEB733
:10290000D790D85CDFB7D754D8FED7E8D7B7D75F12
:10291000D801D618FCF3D7A1D9B8D910E0ADD95F4A
:10292000D8C1D809D60898DEB7D75FD801D6E80352
:10293000CFD854D835D628D658DFB7D709D61054B3
:10294000D8A1D925D810D8CCE835D628D6B4DEB74A
```

FMQ-13 Object Code, "Digital Wind Sensor Application Program" CPIN 83M-FMQ13-F002-00A 31 May 1990

```
:10295000D736E97CD809D604DAD8B8DEB7D736E955
:102960007CD809D604DAD8B4DEA1D9CCDE09D60CDD
:1029700054D8ADD9FED7FCDEC0D754D9B4DEC0D709
:1029800054D9B8DEC0D735D628D6D0DEB7D754D87C
:1029900035D628D630DEB7D701D600013FD727D9A4
:1029A000EDD63201D600044BE944E010D8C1D85C22
:1029B000DFB7D765E11BE1FBE8FED701D684EA016A
:1029C000D6B01D3FD9EDD609ACDEC0D7E5E1E3D6E0
:1029D0000408D8F9E135D628D609D61ACFD854DF5D
:1029E000B7D754D835D628D6FED7FED701D6881308
:1029F00001D668423FD9EDD61201D6E803A4D80922
:102A0000D6055FD854D910D8E3D635FED701D6887D
:102A10001398D7EDD61FFED788D7EDD60954D958CD
:102A2000D95FD8E3D60254D901D6881354D9ECDC47
:102A300054D9E3D60B01D6803E5FD854D909D60BC2
:102A400035D628D6FED7E8D7D7E9E8D7E6E990D833
:102A5000F3D754D8A1D9B8D914E0ADD9FED7E8D767
:102A60005FD801D6E803D9DF3D754D9CCDC01D641
:102A700050C354D9CCDCB8D9DCDE31D809D604CF68
:102A8000D854D8ADD901D6001001D6401FD9DD09E0
:102A9000D60354D9CCDC09D60557DD31D809D60484
:102AA000CFD858DFB7D754D8A1D925D810D8E1E866
:102AB000F3D790D8BCDE54D8C0D735D628D6ACDEF4
:102AC000B7D788E909D60454D9F9D631D842EA2BC8
:102AD000DA08D7F7D4DEC0D735D628D6BCDEFED785
:102AE000E8D709D60654D8B7D731D809D60454D870
:102AF000B7D731D87CD8B7D7F3D7B7D735D628D6FC
:102B000092E9BCEADAEAA1CE482E835D628D6C4DFCA
:102B1000B7D764DFB7D765E11BE198DEB7D7FED73B
:102B200098DFC0D790D864DFB7D754D8A1D9A0DF39
:102B3000ADD935D628D6BCDFB7D768DFB7D765E1C2
:102B40001BE198DEB7D7FED794DFC0D790D868DFF7
:102B5000B7D754D8A1D99CDFADD935D628D694DFC4
:102B6000B7D773D864DFB7D7B7D7FED7E8D7CFD8F2
:102B700098DFB7D773D890D854D854D8FED7A1D9F6
:102B8000A4DFADD9F3D754D8A1D9A8DFADD935D6B4
:102B900028D6E8D7A4DFB7D710D8A0DFB7D75FD83B
:102BA000FED7E8D7A4DFA1D95FD8A0DFA1D95FD82D
:102BB000FED7E8D7C1D854D8FED7A8DFFED7B7D7FD
:102BC00010D8A1D95FD8EAD9C1D854D810D85FD8C5
:102BD000F3D79CDFB7D75FD89CDF8A1D95FD8C1D826
:102BE00054D835D628D604DFB7D736E900DFB7D7B3
:102BF00036E954D904DFC0D754D900DFC0D735D661
:102C000028D6FED7E8D7A1D925D845DDF3D7ADD949
:102C100035D628D6E4EB44E0FCDEB7D7C1D8B8DF20
:102C200000EC44E0FCDEB7D7C1D8C0DF00EC35D6FD
:102C300028D601D600200CEB34EB6CDFB7D75CEB69
:102C400090EBFED701D67C1501D604293FD9EDD6ED
:102C500009D8DEC0D7E5E1E3D60408D8F9E135D6D6
:102C600028D628DFD3D776D9EDD60573D8E3D64555
:102C700009D607B8DFA1D988D9ACDFADD909D604D8
:102C800045DD09D603C8E1BCDFC0D7C0DFA1D9B894
:102C9000D9B0DFADD909D60445DD09D603C8E1C4F2
:102CA000DFC0D754D954D9B8DFADD954D954D9C01D
:102CB0000DFADD930EC28DFDFD735D628D630DEB708
:102CC0000D701D600013FD727D934DFD3D7B4DFDF10
:102CD0000D7EDD622CCDFD3D7EDD60512ECE3D61450
:102CE00001D6F000CCDFDFD754D904DFC0D754D9E8
:102CF00000DFC0D760EC35D628D6F4DEB7D784DF46
:102D0000B7D765E11BE198DEB7D7FED7E8DFC0D7BC
:102D100090D884DFB7D754D8A1D9F0DFADD935D654
:102D200028D6F0DEB7D788DFB7D765E11BE198DE9C
```

```
:102D3000B7D7FED7ECDFC0D790D888DFB7D754D845
:102D4000A1D9F4DFADD935D628D6ECDFB7D773D803
:102D500084DFB7D7B7D7FED7E8D7CFD8E8DFB7D764
:102D600073D890D854D854D8FED7A1D9F8DFADD9AC
:102D70003D754D8A1D9FCDFADD935D628D6E8D7BA
:102D8000F8DFB7D710D8F0DFB7D75FD8FED7E8D7CE
:102D9000F8DFA1D95FD8F0DFA1D95FD8FED7E8D797
:102DA000C1D854D8FED7FCDFDFD7B7D710D8A1D9E9
:102DB0005FD8EAD9C1D854D810D85FD8F3D7F4DF98
:102DC000B7D75FD8F4DFA1D95FD8C1D854D835D6EA
:102DD00028D640DFFED7D3D76AD809D605E6D81063
:102DE000D8DFD735D628D6B8D91CE0ADD9E8DE4033
:102DF000DFD3D709D604CFD8FED7E8D754D8ADD97A
:102E0000B8D920E0ADD9ECDEF3D754D8ADD9D0EDA8
:102E100035D628D6F9D60CDB8D90BE6EAD90CDDC3
:102E2000B8D90BE664E6FCE72BDA35D628D6FED710
:102E30004BDD25D8FED74BDDCCDCCE735D628D60A
:102E4000B8D9F9D6CED9B8D9F9D612EE09D60254E6
:102E5000D9CCDC09D60457DDEAD909D60457DDEA16
:102E6000D909D60457DD10D839E7F4DEC0D7F0DE33
:102E7000C0D735D628D6B8D9DFE64BDD58D97CDDAA
:102E800001D60008B1DDF9D612E74BDD58D97CDD5B
:102E900001D60008B1DDEAD935D628D698DEB7D7F5
:102EA00090D888E054D8A1D9F9D683D95FD809D66B
:102EB000F6EAD95FD8C1D898DEB7D709D6F6CFD809
:102EC00001D6060454D854D835D628D6ACDEB7D7A8
:102ED00088E065E11BE19AEEFED708DFC0D701D696
:102EE000E80310D8C1D8FED730E0B7D7A0D7EDD6C9
:102EF0000901D60010F9E1E3D60601D60010E5E19C
:102F000010D835D628D6FED704E0C0D7F0DEB7D724
:102F100000E0B7D75FD82CE0B7D7D8DEB7D7C1D895
:102F200035D628D6F9D631D8A1D9CCDC09D6041E9D
:102F3000D7F435D628D6FED788D7EDD60909D6ECF2
:102F400009D6FFE3D60509D61454D9CCDC09D62816
:102F500057DD35D628D654D954D9E8DE09D614051C
:102F6000E122EF34EFA0DEC0D754D954D9ECDE090A
:102F7000D61405E122EF34EFA4DEC0D735D628D62B
:102F8000F8EC20ED8CDFB7D748ED7CED00E0C0D742
:102F900035D628D690DFB7D748EDF4DEB7D7FED7C1
:102FA000F0DEB7D77CED5FD8FED788D7EDD6090124
:102FB000D6004054D8E3D611FED701D60040A0D7A2
:102FC000EDD60601D600405FD835D628D6F9D631E1
:102FD000D8B7D745DD09D6021ED7F435D628D65442
:102FE000D954D9E0DE09D60805E1CBEF10DFADD921
:102FF00054D954D9E4DE09D60805E1CBEF14DFAD8E
:10300000D935D628D6A8E058D944DFDFD7DDEF1070
:10301000DFA1D914DFA1D93EEE7EEF2BDA92EF2BA0
:10302000DA04EFCACE74EEE5ED54EF6CE82BDABB90
:10303000ECAFE035D637F05F5A58F7F750AD97FF51
:103040002543F0595805010083D100D1F9D1D8505A
:10305000AD97FF2528D640DEB7D776D9EDD61873C1
:10306000D8FED740DEC0D760D7EDD60B54D93CDEB2
:10307000DFD709D610E5E135D628D609D60454D9CC
:10308000F9D6A8DE31D890D854D8B7D736E990D839
:10309000FED74BDDCDE31D809D604CFD854D8A119
:1030A000D941F0FED7E8D754D941F045DDF3D73503
:1030B000F0C0DE31D890D854D8C0D708D7C435D6A0
:1030C00028D6C0DEFED7A1D910D85FD8C4DEC0D7BD
:1030D00009D60454D8A1D910D85FD8C8DEC0D735D6
:1030E000D628D6A1D9B8D954D825D825D898D7ED7F
:1030F000D60211D935D628D63C9FD3D790D854D8AC
:10310000C0D735D628D6FED7E8D798DEB7D790D81F
```

FMQ-13 Object Code, "Digital Wind Sensor Application Program" CPIN 83M-FMQ13-F002-00A 31 May 1990

```
:1031100054D8FED7E8D7B7D75FD831D8A1D95FD870
:10312000F3D7F3D7B7D754D809D60454D8FED7E885
:10313000D7A1D95FD810D8C1D8F3D7B7D754D835CD
:10314000D628D674DFB7D765E11BE174DFB7D704A3
:10315000F135D628D67CDFB7D765E11BE17CDFB738
:10316000D704F135D628D63CDFFED7D3D76AD809A5
:10317000D6033FD710D8DFD735D628D6FED7E8D725
:103180003CDFD3D790D854D8B7D71AD85FD81BD93B
:1031900048E0A0D7EDD6189AD854D954D931D809D7-
:1031A000D60805E1CBEF25D845DD09D60557DDF377
:1031B000D735D628D601D6000209D60454D9F9D677
:1031C000A8DE31D890D854D8B7D701D6E80301D6B5
:1031D000803E3FD908D7E93FD73FD73FD7EDD60B41
:1031E000E5E154D9D0DFDFD7E3D622D0DFFED7D355
:1031F000D76AD809D61229D9FED725D8DFD709D65C
:103200001270D7EDD605F9E1E3D60208D835D628F5
:10321000D6FED71BD93CE0B7D7A0D7EDD64ADCDF26
:10322000D3D7EDD624FED71BD940E0B7D7A0D7ED32
:10323000D61408D8C0DE7CD8E1F053F1E4DFB7D76C
:1032400001D60040C1D8E3D61CFED7C0DE7CD8E151
:10325000F053F101D6004010D8C1D8E4DFC0D758F0
:10326000D9DCDFDFD7E3D60654D9DCDFDFD735D6AC
:1032700028D6FED71BD934E0B7D7A0D7EDD646D88D
:10328000DFD3D7EDD622FED71BD938E0B7D7A0D7EA
:10329000EDD61208D8C0DEE1F041F1E0DFB7D7018A
:1032A000D60040C1D8E3D61AFED7C0DEE1F041F126
:1032B00001D6004010D8C1D8E0DFC0D758D9D8DF38
:1032C000DFD7E3D60654D9D8DFDFD735D628D6B333
:1032D000F179F0C0F0C4DEB7D741F1E0DE7AF1F663
:1032E000FOC8DEB7D753F1E4DE7AF1F6F065F1A865
:1032F0000DECCDE09D6240FD958D944DEDFD735D647
:1033000028D679F0C0F0C4DEB7D741F1E0DEF6F0A0
:10331000C8DEB7D753F1E4DEF6F035D628D62BDA7F
:1033200080DED3D7EDD6F754D980DEDFD700F38027
:10333000DED3D7EDD63754D980DEDFD7CDF22BDA06
:1033400034DFD3D7FED7B4DFDFD709D6033FD7FEAC
:10335000D760D7EDD60908D803F02BDAE3D60C09ED
:10336000D60370D7EDD604FEEA2BDA54F02BDAE35D
:10337000D6BD35D628D611DAB7D7D3D758D911DA72
:10338000C9D70EDAB7D773D80EDAC0D774E0DFD753
:103390003EDA35D628D63EDA35D69CF35888060272
:1033A00040AD97FF25A7F38A0603402502003402AB
:1033B00050AD97FF2528D62BDAAF360D7EDD6F7C9
:1033C00009D6259AF309D60354D9F9D601D6FF00B8
:1033D000D7DA31D854D8DFD708D7F1D7DA09D603EE
:1033E00086DA35D628D609D6249AF32BDAAF360E7
:1033F000D7EDD6F709D6649AF335D628D609D64044
:1034000064DED3D709D61070D7EDD60BE5E154D9D9
:1034100070DEDFD7E3D62070DED3D76AD809D603B3
:1034200029D9FED770DEDFD709D60370D7EDD605D0
:10343000F9E1E3D602E5E154D964DEDFD735D641C0
:10344000F45FB80000BA00008A1552B908005A5259
:1034500033D083E2017504D1E8EB07D1EAD1D83546
:1034600001205AD0EA52E2E65A475A4A5275D95ACE
:1034700050AD97FF2528D6B5F350DE6AD85FD8FE49
:10348000D786D874DEDFD709D60354D9F9D650DEF3
:103490006AD81AD886DA08D7F508D8FBF3E4F335EA
:1034A000D628D61AD8DFD76AD835D628D61AD8C0A3
:1034B000D77CD835D628D6B5F324E009D60786DAE6
:1034C00050DE6AD858D9A1F409D602A1F409D6036E
:1034D000A1F4A8DE09D60C05E1F9D631D8B7D7ABEF
:1034E000F409D6021ED7F4D8DEB7D7ABF418DFB78D
```

```
:1034F000D7ABF430DEB7D7ABF41CDFB7D7ABF4C42F
:10350000DFB7D709D602D8E1ABF4C0DE09D60805BB
:10351000E1F9D631D8B7D7ABF409D6021ED7F409F2
:10352000D60DA1F409D60AA1F450DE6AD85FD850AE
:10353000DE6AD810D886DAE4F335D628D609D60361
:1035400054D9F9D654DE31D890D854D8B7D701D64B
:103550000001A4D810D825D809D60254D9F9D61A12
:10356000D8A1F410D809D61070D7EDD60509D61019
:10357000A1F408D7EA2BDA08D7CA35D628D650DE08
:103580006AD809D610A1F409D602A1F410D8A1F482
:1035900060DED3D7A1F43BF509D610A1F409D60318
:1035A000A1F4FED750DE09D60354D85FD850DE0907
:1035B000D60354D83FF41AD8C0D77CD801D6FF0020
:1035C000A1F475F435D628D67CD810D86AD8D3D7CC
:1035D00060DED7D7A0DEB7D754DEC0D7A4DEB7D712
:1035E00054DE7CD8C0D730DEB7D754DE09D60454B9
:1035F000D8C0D754D930DE7CD8C0D735D628D67CB1
:10360000D810D86AD8D3D760DED3D770D727D9EDF2
:10361000D60C86D858DEB7D76AD810D8C6F535D6B6
:1036200028D658DEB7D76AD8FED7D3D7FED709D663
:10363000033FD709D60270D7EDD605FDF5E3D602D4
:10364000C6F57CF535D628D6B5F324E009D64086F4
:10365000DAE4F335D628D658DEB7D76AD8D3D7FE02
:10366000D709D6303FD709D610DAD834DED3D709F8
:10367000D6033FD770D7EDD640FED709D6073FD740
:10368000FED758D970D710D809D60270D74AD72795
:10369000D9EDD62209D60F3FD7FED709D60770D766
:1036A000EDD60708D8B5F4E3D60A09D60670D7EDEB
:1036B000D60246F654D9E3D60227D935D628D65CA9
:1036C000DEB7D758DEB7D7B8D95FD810D86AD83F99
:1036D000F42BDA10D86AD8B7D770D735D628D62BBE
:1036E000DA54D95CDEC0D750DE09D61054D810D8D1
:1036F000F9D631D8D3D709D61070D7EDD62331D823
:103700006AD8D3D709D60370D7EDD60F31D86AD887
:103710005CDEC0D758D995D9E3D60309D602E3D6E3
:10372000258D91ED7CC5CDEB7D72BDA35D628D6CF
:103730002BDA54D958DEC0D750DE09D60854D81039
:10374000D8F9D631D8D3D709D61070D7EDD62331D2
:10375000D86AD8D3D709D60270D7EDD60F31D86A38
:10376000D858DEC0D758D995D9E3D60309D602E395
:10377000D60258D91ED7CC58DEB7D72BDA35D62883
:10378000D650DE6AD82EF7FED7EDD60E6AD8DDF613
:10379000EDD605BDF6E3D60254D935D628D67FF747
:1037A000EDD60755F6EDD60220F635D628D650DEF2
:1037B00009D62867D950DE6AD809D6279CDA50DEA8
:1037C000D3D7EDD6029CF72BDAE3D6E235D65B5998
:1037D00058CFD4F7B80000CFFF06A401CF50515303
:1037E000BBFF068A060040F6471480791C877F16C7
:1037F000880547877F16FF4718FF47147509C606D7
:10380000AD0000C707FFD7EB10F606E20001740910
:103810008806E000C606E20000EBB38B47140BC03D
:103820007E24877716AC87771688060040FF4F14F2
:103830007506C707FFD7EB0E8A47143806E40075F4
:1038400005C606E20001C3505153BBFF06BA0601BC
:10385000408AC8A8207403E8C1FFF6C11074238A07
:10386000060340240234027515F606E10002740EC8
:10387000C606AD0001C747140000C707FFD788067A
:10388000E100E949FF5051539C8A06AD01402507EC
:1038900008806AD01C606AC0101C606AE0101C72F
:1038A00006A60100009DE925FF28D6E4EB09D6080D
:1038B0004BDB8DFADD94BE209D6084BDDC0DFAD3B
:1038C000D935D628D620DFD3D709D60A98D7EDD652
```

FMQ-13 Object Code. "Digital Wind Sensor Application Program" CPIN 83M-FMQ13-F002-00A 31 May 1990

```
:1038D0003654D9B4DEC0D754D9B8DEC0D709D6041F
:1038E00054D9F9D609D604F8E4B4DEC9D709D60606
:1038F000F8E4B8DEC9D708D7EBB4DEB7D7B8DEB77F
:10390000D70CE0ADD992E935D628D609D62881E181
:1039100035D628D654D942EA01D6102754D858D9DA
:1039200042EA08D8BCDEA1D935D628D6ACDEB7D756
:10393000FED701D6A8E498D7EDD60408D854D935D7
:10394000D628D620DFD3D709D60A98D7EDD641544A
:10395000D9B4DEC0D754D9B8DEC0D754D900DFC03F
:10396000D754D904DFC0D709D60454D9F9D692E583
:1039700008D7FB00DFB7D704DFB7D70CE0ADD9FE1F
:10398000EA42E401D6102754D9F9D64BE208D7FB16
:1039900035D628D6F9D631D831D8B8D91CE442E486
:1039A00009D67854D9F9D64BE208D7FB08D7E735C2
:1039B000D628D6C3F82AF94BE212F9FED7FED7B8BB
:1039C000D91CE442E409F992F94BE241F9A9F8352E
:1039D000D6A1D6FF062EEAFF05000000007E060AEB
:1039E0000076F396F3A1D6FF052EEAFF040000004F
:1039F00000BE050A00A1D6FF042EEAFF0300000066
:103A000000BE040A00A1D6FF032EEAFF0600000054
:103A100000BE030A00A1D6D8F700000DDF70000477A
:103A2000F80000DF7000028D6FED7B7D754D80934
:103A3000D60454D8FED760DFC0D7FED7B7D754D846
:103A400009D60454D8FED764DFC0D7FED7B7D75401
:103A5000D809D60454D8FED768DFC0D7FED7B7D769
:103A600054D809D60454D8FED76CDFC0D764DFB76A
:103A7000D7B7D768DFB7D7B7D7CFD89AD87CD854BD
:103A8000D8FED770DFC0D7FED7B7D754D809D60431
:103A900054D8FED774DFC0D7FED7B7D754D809D6CD
:103AA0000454D8FED778DFC0D7FED7B7D754D8098B
:103AB000D60454D8FED77CDFC0D7FED7B7D754D8AA
:103AC00009D60454D8FED780DFC0D7FED7B7D75465
:103AD000D809D60454D8FED784DFC0D7FED7B7D7CD
:103AE00054D809D60454D8FED788DFC0D7FED7B73C
:103AF000D754D809D60454D8FED78CDFC0D784DF7A
:103B0000B7D7B7D788DFB7D7B7D7CFD89AD87CD8A9
:103B100054D890DFC0D735D628D624E009D67054C3
:103B2000D8FED754DFC0D709D66A54D8FED758DF9D
:103B3000C0D709D61654D8FED75CDFC0D727FA35D0
:103B4000D628D628E0A1D901D6000145DD01D6004E
:103B50000257DD35D657FBC606036034C606036040
:103B600056C6060360B0588806006088260060C606
:103B700006026000C606026000C606016050AD97EE
:103B8000FF2528D609D63C78E0DFD701D6A20080F1
:103B9000E0DFD709D6647CE0DFD774E0D3D708D85C
:103BA00074E0D3D708D878E0D3D708D835D6B0FB9F
:103BB000FBAD97FP25B7FB800EEB00408A06EB00BC
:103BC000088060080AD97FF2528D609D6103CDED3A5
:103BD000D76AD801D6FF0029D9FED73CDEDFD70946
:103BE000D60598D7EDD605E5E1E3D602F9E101D691
:103BF000600940DEC0D735D6FAFB8A060280243F32
:103C00008806A400AD97FF2528D682FB41FB55FB13
:103C10004BE235D628D62CDE01D62C01B8D967D98F
:103C20004BE254D801D62C0167D94BE235D628D6C1
:103C300001D61027D0DEC0D701D61027D8DEC0D7D6
:103C400001D64871B0DEC0D701D648710CDFC0D7AD
:103C500001D6D007ACDEC0D701D6393038DEC0D7A8
:103C600058D9D8DFDFD758D9DCDFDFD701D60040FD
:103C7000E0DFC0D701D60040E4DFC0D754D990DEE2
:103C8000DFD754D928DFDFD709D6022CDFDFD7549E
:103C9000D9CCDFDFD735D628D6E5F94EDAE0E535E1
:103CA000D628D6F5F94EDA1CF335D628D605FA4EC5
```

FMQ-13 Object Code, "Digital Wind Sensor Application Program" CPIN 83M-FMQ13-F002-00A 31 May 1990

```
:103CB000DA07E435D628D6A1FCABFC35D628D624C5
:103CC000DFD3D7EDD63020DFFED7D3D76AD8FED7E3
:103CD00025D8DFD709D610A0D7EDD61501D600011B
:103CE000F9E101D62EEAE5F9F9D940DAC0D7E3D6F1
:103CF0000297FCE3D60297FC35D628D608FC38DEBE
:103D0000B7D701D6393070D727D9EDD60214FC2E9B
:103D1000FCC8FBF8FB18FBBDPC4BE2B1F9B5FCAEEF
:103D2000FBB5FBACF735D64BE215FA01D68000099E
:103D3000D610F3D854D909D602C0D701D6D4F75437
:103D4000D9C0D754D909D60AC0D701D685F809D623
:103D500008C0D7D1F97CD8D1F9F9D940DAFED70912
:103D6000D63C67D909D61CF3D8E5F97CD8E5F9F932
:103D7000D940DAFED709D63C67D909D60CF3D8F575
:103D8000F97CD8F5F9F9D940DAFED709D63C67D9DC
:103D900009D60CF3D805FA7CD805FAF9D940DAFE31
:103DA000D709D63C67D909D60CF3D8FAFC35D6B872
:103DB00000008EC08ED08ED8BE27FD8B1ED3F98B0F
:103DC000EB8B26DDF9AD97FF25FFFFFFFFFFFFF20
:103DD0000FFFFFFF3
:103DE0000FFFFFFF3
:103DF0000FFFFFFF3
:103E0000FFFFFFF3
:103E10000FFFFFFF92
:103E20000FFFFFFFB2
:103E30000FFFFFFF92
:103E40000FFFFFFF82
:103E50000FFFFFFF72
:103E60000FFFFFFF62
:103E70000FFFFFFF52
:103E80000FFFFFFF42
:103E90000FFFFFFF32
:103EA0000FFFFFFF22
:103EB0000FFFFFFF12
:103EC0000FFFFFFF02
:103ED0000FFFFFFF2
:103EE0000FFFFFFF2
:103EF0000FFFFFFF2
:103F0000FFFFFFF1
:103F10000FFFFFFF1
:103F20000FFFFFFF1
:103F30000FFFFFFF1
:103F40000FFFFFFF81
:103F50000FFFFFFF71
:103F60000FFFFFFF61
:103F70000FFFFFFF51
:103F80000FFFFFFF41
:103F90000FFFFFFF31
:103FA0000FFFFFFF21
:103FB0000FFFFFFF11
:103FC0000FFFFFFF01
:103FD0000FFFFFFF1
:103FE0000FFFFFFF1
:103FF0000EAFFD0000FFFFFFF36
:00000001
```

321 LIST

```
0 ( SENSOR MEMORY ALLOCATION )
1
2 HEX 1F1F WIDTH !           ( SOURCE COMPILE W/ 31 BYTE NAME )
3
4 FFFF EQU /PROM            ( LAST ADD. OF EPROM [FFFF] )
5
6 0000 EQU ORAM             ( 1ST ADD. OF RAM [0000] )
7 7FF EQU TOPRAM             ( LAST ADD. OF RAM [7FF] )
8
9 0000 EQU ZERO              ( SEGMENT REGISTERS OFFSET )
10
11 2 DICTIONARY             ( SYSTEM TARGET COMPILATION DICT. )
12
13 /PROM 29FF - WINDOW       ( 1ST EPROM COMPIL. ADD. [D600] )
14 A0 ALLOT                  ( 1ST RAM COMPILATION ADD. [A0] )
15 .S
```

322 LIST

```
0 ( SENSOR TASKS RAM ALLOCATION )
1
2 HEX
3 TOPRAM 100 - EQU 'OPERATOR      ( INTER-ASSY TASK AREA [6FF] )
4
5 TOPRAM 200 - EQU 'ACQUISITION    ( DATA ACQ. TASK AREA [5FF] )
6
7 TOPRAM 300 - EQU 'PROCESS        ( PROCESS TASK AREA [4FF] )
8
9 TOPRAM 400 - EQU 'WDOG          ( SYS. TEST TASK AREA [3FF] )
10
11 TOPRAM 4C0 - EQU 'COMMAND       ( *****COMMAND TASK AREA [33F] )
12
13
14 .S
15
```

323 LIST

```
0 ( SENSOR DATA TABLE ALLOCATION )
1
2 HEX 1F1F WIDTH !           ( SOURCE COMPILE W/ 31 BYTE NAME )
3
4 FFFF EQU /PROM            ( LAST ADD. OF EPROM [FFFF] )
5
6 0000 EQU ORAM             ( 1ST ADD. OF RAM [0000] )
7 7FF EQU TOPRAM             ( LAST ADD. OF RAM [7FF] )
8
9 0000 EQU ZERO              ( SEGMENT REGISTERS OFFSET )
10 2 DICTIONARY             ( SYSTEM TARGET COMPILATION DICT. )
11
12 /PROM 3FE1 - WINDOW       ( 1ST EPROM COMPIL. ADD. [C01E] )
13 A0 ALLOT                  ( 1ST RAM COMPILATION ADD. [A0] )
14 DECIMAL 610 617 THRU ( DATA TABLES   )
15 HEX HERE U. THERE U.     FLUSH .S
```

324 LIST

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

325 LIST

0 (McCLELLAN SENSOR LOAD SCREEN 8/8/86)
1 (1ST ART. REV E RELEASE 4/04/89) HEX 1F1F WIDTH ! DECIMAL
2
3 340 353 THRU (VARIABLES & CONSTANTS)
4 354 LOAD (***** NOPS FOR TESTING)
5 355 360 THRU (SYSTEM ROUTINES)
6
7 430 443 THRU (CPU, RAM, ROM TESTS & BIT ROUTINES)
8
9 365 374 THRU (D/A & A/D & DATA ACQUISITION ROUTINES)
10
11 380 427 THRU (DATA PROCESSING ROUTINES)
12
13 326 LOAD (LOAD SCREEN CONT.)
14
15 .S

326 LIST

0 (McCLELLAN SENSOR LOAD SCREEN 8/8/86)
1 428 LOAD (APPLICATION PROCESSING)
2 452 471 THRU (INTER-ASSEMBLY ROUTINE)
3
4 480 487 THRU (INTERRUPT HANDLERS)
5
6 499 504 THRU (ELEMENT POWER UP ROUTINE)
7
8 509 511 THRU (TASK TABLE & SYSTEM INITILIZATION)
9
10 512 520 THRU (SYSTEM & TASK INITILIZATION & START-UP)
11
12 525 526 THRU (SYSTEM 'START' ROUTINE)
13 .S
14
15

327 LIST

0 (MALLELLAN SENSOR LOAD SCREEN 8/8/86)
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

328 LIST

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

329 LIST

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

339 LIST

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

340 LIST

0 (SENSOR SYSTEM VARIABLES 6/27/86)
1
2 THERE CONSTANT RAMSTART
3
4 HEX 1F1F WIDTH ! DECIMAL
5
6 VARIABLE SENSORSTAT 2 ALLOT (SENSOR STATUS & ACK WORD)
7 CVARIABLE CONFIG (SENSOR CONFIGURATION)
8 VARIABLE SETUPFLAG (SENSOR SETUP FLAG)
9 CVARIABLE RESETCTR (SYSTEM STARTUPS COUNTER)
10 VARIABLE RESETIMER (10 MIN RESET TIMER)
11
12 CVARIABLE PRINTFLAG (OUTPUT DATA FLAG)
13 CVARIABLE XFLAG (X-ON/X-OFF FLAG)
14 CVARIABLE SCROLL (SCROLL FLAG)
15 .S

341 LIST

0 (SENSOR INTER-ASSY TASK VARIABLES)
1
2 VARIABLE COMMBUFF 38 ALLOT (COMMUNICATION BUFFER)
3 VARIABLE POLLTEMP 4 ALLOT (TEMP BUFFER FOR POLL RESP DATA)
4 VARIABLE STXPTR (PTR TO STX CHAR IN COMMBUFF)
5 VARIABLE ETXPTR (PTR TO ETX CHAR IN COMMBUFF)
6 CVARIABLE POLLCNT (LAST POLL REQUEST #)
7 CVARIABLE CHARBUFF (TEMP FOR LOOPBACK TEST CHAR)
8 CVARIABLE CDFLAG (CARRIER DETECT STATUS FLAG)
9 CVARIABLE TESTFLAG (TEST LOOPBACK CHAR FLAG)
10 CVARIABLE UARTERR (XMITTER LOOPBACK ERROR CTR)
11 CVARIABLE CHARNCT (# OF CHARS TO XMIT)
12
13
14 .S
15

342 LIST

```

0 ( SENSOR PROCESS TASK VARIABLES )
1
2 CVARIABLE d/a.index ( INDEX FOR D/As)
3 CVARIABLE DATA.ERROR ( INPUT DATA ERROR FLAGS)
4 CVARIABLE PROCESSFLAG ( PROCESS ENABLE FLAG)
5 CVARIABLE chan ( channel # for sample_elements)
6 CVARIABLE loopcnt ( loop count for sample_elements)
7 CVARIABLE a/d# ( pointer to current a/d channel)
8 CVARIABLE AD/CONREG ( A/D SOFTWARE REGISTER)
9
10 VARIABLE INDEX ( INDEX FOR BINARY SEARCH ROUTINE)
11 VARIABLE BOTTOM ( BOTTOM POINTER FOR BINARY SEARCH)
12 VARIABLE TP ( TOP POINTER FOR BINARY SEARCH) .S
13 VARIABLE XBAR ( 5-SECOND X AVERAGE )
14 VARIABLE YBAR ( 5-SECOND Y AVERAGE )
15

```

343 LIST

```

0 ( SENSOR PROCESS TASK VARIABLES )
1
2 VARIABLE ELEMENTS 6 ALLOT ( SENSOR ELEMENT READINGS)
3 VARIABLE TEMPERATURE ( TEMPERATURE VALUE)
4 VARIABLE FREQUENCY ( AIR DENSITY SENSOR FREQUENCY)
5 VARIABLE TEMPVOLT1 ( TEMPERATURE SENSOR VOLTAGE)
6 VARIABLE VOLTS ( VOLTAGE READING OF +5 SUPPLY)
7 VARIABLE D/AOUTBUF 6 ALLOT ( D/A OUTPUT VALUE TO HEAT ELEMENTS)
8 VARIABLE ELEMENTPOWER 6 ALLOT ( ARRAY OF LATEST ELEMENT POWERS)
9 VARIABLE XPOWER ( POWER IN X ELEMENT PAIR )
10 VARIABLE YPOWER ( POWER IN Y ELEMENT PAIR )
11 VARIABLE HOLDBUF 34 ALLOT ( DUMP OUTPUT BUFFER)
12 VARIABLE OVERHEAT ( OVERHEAT OFFSET VALUE)
13 VARIABLE OVERHEATSETPT ( DESIRED OVERHEAT VALUE)
14 VARIABLE PRESSURE ( ATMOSHEREIC PRESSURE VALUE)
15 VARIABLE RESISTANCE 14 ALLOT ( ARRAY OF LATEST ELEMENT RES) .S

```

344 LIST

```

0 ( SENSOR PROCESS TASK VARIABLES )
1
2 VARIABLE XRAWQ 6 ALLOT ( QUE FOR RAW X VELOCITIES)
3 VARIABLE YRAWQ 6 ALLOT ( QUE FOR RAW Y VELOCITIES )
4 VARIABLE X5SECQ 18 ALLOT ( QUE FOR 5 COMP. X VELOCITIES)
5 VARIABLE Y5SECQ 18 ALLOT ( QUE FOR 5 COMP. Y VELOCITIES )
6 VARIABLE SPEED ( VELOCITY MAGNITUDE )
7 VARIABLE DIRECTION ( VELOCITY DIRECTION )
8 VARIABLE OSVOLTS ( 5 VOLT READING FOR A/D TEST )
9 VARIABLE CSVOLTS ( CORRECTED 5 VOLT READING )
10 VARIABLE VBARO ( PRESSURE VOLTAGE READING )
11 VARIABLE VBTEMP ( PRESSURE TEMP. VOLTAGE READING )
12 VARIABLE TEMPCO ( TEMPERATURE COMPENSATION )
13 VARIABLE FREQUENCY1 ( LASTEST FREQUENCY READ )
14 2VARIABLE XUNCOMP ( UNCOMPENSATED 1 SEC X VELOCITY)
15 2VARIABLE YUNCOMP .S ( UNCOMPENSATED 1 SEC Y VELOCITY)

```

345 LIST

```

0 ( SENSOR SYSTEM TEST TASK VARIABLES )
1 VARIABLE SPURCNT          ( SPURIOUS INTERRUPT CONUTER)
2 VARIABLE NMICHKCNT         ( FLAG TO CHECK NMI )
3 CVARIABLE A/DFAILCTR      ( # OF A/D FAILURES)
4 CVARIABLE A/DSTATE         ( A/D FLAG SET DURING CONVERSION)
5 CVARIABLE SECCNT           ( SECONDS COUNTER FOR PROCESSING )
6 CVARIABLE 2SECCNT          ( SECONDS COUNTER FOR PROCESSING )
7 CVARIABLE ACQUIRE          ( FLAG TO START DATA ACQUISITION )
8 CVARIABLE PHASE             ( MODULO 8 CTR FOR 8 1/4 SECONDS )
9 CVARIABLE RAMFLAG           ( FLAG TO SYNC START OF RAM TEST )
10 CVARIABLE RAWTAIL          ( POINTER TO END OF RAW QUE )
11 CVARIABLE SSECTAIL         ( POINTER TO END OF 5 SEC QUE)
12 CVARIABLE DACFLAG          ( FLAG TO LOAD DACS)
13 CVARIABLE LASTDAC          ( LAST DAC OUTPUT VALUE)
14 CVARIABLE VERROR            ( VOLTAGE ERROR COUNT)
15 CVARIABLE FREQFLAG          ( FLAG TO READ FREQUENCY) .S

```

346 LIST

```

0 ( SENSOR PROCESS TASK VARIABLES   POINTERS TO LOOKUP TABLES )
1
2 VARIABLE ELEMENTOHMS          ( ELEMENT RESISTANCE DATA )
3 VARIABLE GAINS/OFFSETS        ( A/D & D/A GAIN & OFFSET DATA )
4 VARIABLE THERMIS              ( THERMISTER DATA )
5 VARIABLE FREQTABLE             ( FREQUENCY TO PRESSURE TABLE )
6 VARIABLE BTEMPDEX              ( PRESSURE COMP. TEMP. INDEX TABLE)
7 VARIABLE BARODEX              ( PRESSURE COMP. PRESSURES INDEX TABLE)
8 VARIABLE BAROCOMP              ( PRESSURE COMPENSATION DATA TABLE)
9 VARIABLE BAROTEMP              ( PRESSURE SENSOR TEMP. DATA)
10 VARIABLE PDX                  ( X-AXIS POWER DIFFERENCE DATA )
11 VARIABLE RAWX                 ( X-AXIS RAW VELOCITY DATA )
12 VARIABLE PDY                  ( Y-AXIS POWER DIFFERENCE DATA )
13 VARIABLE RAWY                 ( Y-AXIS RAW VELOCITY DATA )
14 .S
15

```

347 LIST

```

0 ( SENSOR PROCESS TASK VARIABLES   POINTERS TO LOOKUP TABLES )
1
2 VARIABLE ANGLEDEX             ( MAG. COMP. ANGLE INDEX TABLE)
3 VARIABLE MAGDEX               ( MAG. COMP. MAG. INDEX TABLE)
4 VARIABLE MAGCOMP               ( MAG. COMP. DATA TABLE)
5 VARIABLE ANGLECOMP              ( ANGLE COMPENSATION TABLE)
6
7 VARIABLE BBOTTOM                ( INDEX FROM BARO. TABLE )
8 VARIABLE BTBOTTOM               ( INDEX FROM BARO. TEMP. TABLE )
9 2VARIABLE BPAIR                 ( BARO. COMPENSATION PAIR )
10 2VARIABLE BTPAIR                ( BARO. TEMP. COMPENSATION PAIR )
11 2VARIABLE LBCPAIR               ( LOWER BARO. COMPENSATION PAIR )
12 2VARIABLE UBCPAIR               ( UPPER BARO. COMPENSATION PAIR )
13
14
15 .S

```

348 LIST

```
0 ( SENSOR PROCESS TASK VARIABLES )
1
2 2VARIABLE BAROTOT      2VARIABLE BTEMPTOT CVARIABLE PHASE? ( ****)
3 2VARIABLE BAROSUM      ( SUM OF 1 SEC. PRESSURE VOLTAGE READINGS )
4 VARIABLE BAROBAR       ( AVERAGE PRESSURE VOLTAGE READING )
5 2VARIABLE BTEMPSUM     ( SUM OF 1 SEC. PRESSURE TEMP. READINGS )
6 VARIABLE BTEMPBAR      ( AVERAGE PRESSURE TEMP. READING )
7 CVARIABLE HEATFLAG    ( HEATER ACTIVE FLAG)
8 CVARIABLE PTINIT      ( PRESSURE TRANSDUCER INITIALIZATION FLAG )
9 CVARIABLE ELECTR      ( ELEMENT TEST ERROR COUNTER )
10 CVARIABLE FECTR      ( FREQUENCY ERROR COUNTER)
11 CVARIABLE XSFLAG      ( SCALING FLAG FOR X AXIS )
12 CVARIABLE YSFLAG      ( SCALING FLAG FOR Y AXIS )
13 VARIABLE XSCALE       ( XSCALE FACTOR )
14 VARIABLE YSCALE       ( YSCALE FACTOR )
15 .S
```

349 LIST

```
0 ( SENSOR PROCESS TASK VARIABLES )
1
2 VARIABLE ABOTTOM      ( INDEX FROM ANGLE TABLE)
3 VARIABLE MBOTTOM      ( INDEX FROM SPEED TABLE)
4 2VARIABLE APAIR        ( ANGLE COMPENSATION PAIR)
5 2VARIABLE MPAIR        ( MAGNITUDE COMPENSATION PAIR)
6 2VARIABLE LMCPAIR      ( LOWER MAG. COMP. PAIR)
7 2VARIABLE UMCPAIR      ( UPPER MAG. COMP. PAIR)
8 VARIABLE MCMP          ( COMPENSATED MAGNITUDE VALUE)
9 VARIABLE CTHETA        ( COMPENSATED ANGLE)
10 2VARIABLE AGPAIR       2VARIABLE 1STVOLTS ( *** TEST VARIABLE )
11 2VARIABLE TPAIR         ( *** TEST VARIABLE )
12 2VARIABLE RPAIR         ( *** TEST VARIABLE )
13 2VARIABLE FPAIR         ( *** TEST VARIABLE )
14 VARIABLE XCOMP         ( *** TEST COMP. X VEL. )
15 VARIABLE YCOMP         ( *** TEST COMP. Y VEL. ) .S
```

350 LIST

```
0 ( SENSOR HARDWARE DEVICE ADDRESSES      7/7/86)
1 HEX
2 C000 CONSTANT TABLES      ( STARTING ADDRESS OF LOOKUP TABLES)
3      ( TABLE ADDRESSES FOR SENSOR DEPENDENT VARIABLES)
4 C060 CONSTANT XTALFREQ   ( LOCATION OF STORED CRYSTAL FREQ )
5 C064 CONSTANT PRESREF    ( PRESSURE REFERENCE)
6 C066 CONSTANT SPEEDLIMIT ( CALIBRATION RANGE LIMIT )
7 C068 CONSTANT XLOLIMIT   ( LIMITS FOR HIGH VELOCITY SCALING )
8 C06A CONSTANT XHILIMIT
9 C06C CONSTANT YLOLIMIT
10 C06E CONSTANT YHILIMIT
11
12 DECIMAL
13 2048 CONSTANT SVOLTS    ( A/D COUNT FOR 5 VOLT INPUT) .S
14 5120 CONSTANT RAINLIMIT ( DIFFERENCE LIMIT FOR RAIN )
15
```

351 LIST

```

0 ( SENSOR HARDWARE DEVICE ADDRESSES          7/7/86)
1
2 HEX
3 8000 CONSTANT ad/mx/msb ( ADDRESSES FOR A/D-1)
4 8001 CONSTANT ad/lsb
5 8002 CONSTANT ad/csr
6
7 6000 CONSTANT ce0  ( 8254 PROG. COUNTER)
8 6001 CONSTANT ce1
9 6002 CONSTANT ce2
10 6003 CONSTANT ckcsr
11
12 00 CONSTANT da/msb
13 01 CONSTANT da/lsb ( D/A BASE ADDRESS)
14 40 CONSTANT da/csr  ( D/A CONTROL REGISTER) .S
15

```

352 LIST

```

0 ( SENSOR HARDWARE DEVICE ADDRESSES          7/7/86)
1 HEX
2 4000 CONSTANT comm/tr          ( UART REGISTER ADDRESSES)
3 4001 CONSTANT comm/csr
4 4002 CONSTANT comm/mcr
5 4003 CONSTANT comm/msr
6
7 50 CONSTANT watchdog          ( uP WATCHDOG CIRCUIT ADDRESS )
8 .S
9
10
11
12
13
14
15

```

353 LIST

```

0 ( TEMPERATURE COMPENSATION DATA  8/1/87 )
1
2 CREATE TEMPTAB ( TEMPERATURE CORRECTION DATA )
3 14 , -5184 , -2775 , -299 , 2000 , 3897 , 5387 , 6566 ,
4
5
6 .S
7
8
9
10
11
12
13
14
15

```

354 LIST

```
0 ( SENSOR SYSTEM ROUTINES )
1
2      ( ***** REMOVE LATER ***** )
3 CODE no.op NOP NEXT    ( USED FOR DEBUGGING SOFTWARE )
4 CODE no.op1 NOP NEXT   ( USED FOR DEBUGGING SOFTWARE )
5 CODE no.op2 NOP NEXT   ( USED FOR DEBUGGING SOFTWARE )
6 CODE no.op3 NOP NEXT   ( USED FOR DEBUGGING SOFTWARE )
7 CODE no.op4 NOP NEXT   ( USED FOR DEBUGGING SOFTWARE )
8 CODE no.op5 NOP NEXT   ( USED FOR DEBUGGING SOFTWARE )
9 CODE no.op6 NOP NEXT   ( USED FOR DEBUGGING SOFTWARE )
10 ( CODE spno.op4 PHASE 0 MOV B NEXT   ( USED FOR DEBUGGING SW )
11 CODE sp.op 0 POP 1 POP 2 POP 2 PUSH 1 PUSH 0 PUSH NEXT
12 CODE sp.op1 0 POP 1 POP 2 POP 2 PUSH 1 PUSH 0 PUSH NEXT
13 CODE sp.op2 0 POP 1 POP 2 POP 2 PUSH 1 PUSH 0 PUSH NEXT
14
15 .S
```

355 LIST

```
0 ( SENSOR SYSTEM ROUTINES )
1
2 : HEX 16 BASE ! ;      : DECIMAL 10 BASE ! ;
3 : SET ( n, n -- SET UP LIMITS FOR DO LOOP )
4     OVER + SWAP ;
5
6
7 .S
8
9
10
11
12
13
14
15
```

356 LIST

```
0 ( Binary Search )
1
2 : LOOK.UP ( a of table, a of index -- TABLE[index])
3     @ 2* + @ ;
4
5 : BSEARCH ( n, a -- n Search table at "a" for n, sets TP & BOTM)
6     )R
7     BEGIN
8         INDEX @ OVER I INDEX LOOK.UP < NOT      ( n )= table[index] )
9             IF BOTTOM                         ( bottom = index   )
10            ELSE TP THEN                      ( top    = index   )
11            ! BOTTOM @ TP @ OVER - DUP )R 2/ + INDEX !
12            R) 2 ( PAUSE                     ( i = [t-b]/2 + b   )
13            UNTIL R) DROP ;                  ( when top-bottom=1 )
14 .S
15
```

357 LIST

```

0 ( Binary Search Set Up)
1
2 : SEARCH.SETUP ( a -- a SET UP POINTERS FOR BSEARCH)
3     DUP @          ( SAVE ADDRESS)
4     2/ DUP TP !    ( SET TOP POINTER )
5     2/ INDEX !     ( SET INDEX )
6     1 BOTTOM ! ;   ( SET BOTTOM POINTER)
7
8
9
10
11
12
13
14
15

```

358 LIST

```

0 ( SYSTEM ROUTINES                               6/6/87)
1
2 HEX
3 CODE set.csrbit ( n -- set bit n of the sensor cntrl/stat reg)
4     0 POP 0 AD/CONREG OR B ( set bit in psuedo reg)
5     AD/CONREG 0 MOV B      ( setup psuedo reg)
6     0 ad/mx/msb MOV B NEXT ( write to register)
7
8 CODE clear.csrbit ( n -- clear bit n of the control/status reg)
9     0 POP FF #B 0 XOR      ( form clearing mask)
10    0 AD/CONREG AND B     ( clear bit in psuedo reg)
11    AD/CONREG 0 MOV B      ( setup psuedo reg)
12    0 ad/mx/msb MOV B NEXT ( write to register)
13
14 .S
15

```

359 LIST

```

0 ( McCLELLAN SENSOR PRES. SENSOR TOGGLE ROUTINE 2/25/87)
1
2
3 .S
4
5
6
7
8
9
10
11
12
13
14
15

```

360 LIST

```
0 ( MALLELLAN SENSOR  EXTRA MATH ROUTINES      2/25/87)
1
2 CODE UM/ ( d, u -- q  PERFORMS UNSIGNED DIVISION & RETURNS SING)
3           W POP 2 POP 0 POP W DIV 0 PUSH NEXT
4
5 CODE D2/RND ( d -- d/2  DIVIDES DOUBLE BY 2 RETURN ROUNDED SING)
6           1 POP 0 POP 1 # 0 ADD 0 # 1 ADC
7           1 SAR 0 RCR 0 PUSH NEXT
8
9 CODE dn/ ( ud, n -- u  UNSIGNED DIVIDE BY 2 TO THE n )
10          1 POP 0 POP 2 POP
11          BEGIN 0 SAR 2 RCR LOOP 2 PUSH NEXT
12
13 CODE n/ ( u, n -- u  UNSIGNED DIVIDE BY 2 TO THE n )
14          1 POP 0 POP
15          BEGIN 0 SHR LOOP 0 PUSH NEXT .S
```

361 LIST

```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
```

362 LIST

```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
```

363 LIST

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

364 LIST

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

365 LIST

```
0 ( McCLELLAN SENSOR DEVEL. D/A DRIVER    7/28/86)
1 HEX
2 CODE d/astuff ( da4,da3,da2,dai -- )
3     0 #B d/a.index MOV 4 # 1 MOV      ( setup for loop)
4 BEGIN
5     0 POP  da/lsb # 2 MOV          ( setup 0 & 1 )
6     d/a.index 2 ADD B (2) OUT      ( output lsb)
7     0 HI 0 MOV B 2 INC (2) OUT      ( output msb)
8     10 #B d/a.index ADD          ( increment index)
9     LOOP   0 LASTDAC MOV B NEXT      ( save last output)
10
11 CODE dac.load ( load dac using the last output)
12     LASTDAC 0 MOV B           ( load DACs with last value)
13     da/csr OUT 0 #B DACFLAG MOV NEXT ( clear flag)
14 .S
15
```

366 LIST

```
0 ( McCLELLAN SENSOR ELEMENT SAMPLING ROUTINE          7/28/86 )
1 HEX
2 CODE sample_elements ( -- sample 4 chns 4 times )
3   F8 #B AD/CONREG AND AD/CONREG O MOV B    ( setup AD/CONREG)
4   0 ad/mx/msb MOV B                      ( set mux to 0)
5   1 #B chan MOV 0 #B loopcnt MOV        ( setup chan# & loop)
6   04 # 1 MOV ELEMENTS # W MOV          ( setup to clear)
7   BEGIN 0 # W ) MOV W INC W INC LOOP  ( clear 8 byte)
8   0 #B ad/csr MOV AD/CONREG O MOV B  ( start a/d & set mux)
9   chan 0 OR B 0 ad/mx/msb MOV B      ( 0 AD/CONREG MOV B )
10  OF #B A/DSTATE MOV                  ( set flag)
11 LABEL L1  wait 0 #B A/DSTATE MOV     ( wait for conver.)
12  0 #B A/DFAILCTR MOV                ( clear flags)
13  0 0 SUB ad/mx/msb 0 MOV B         ( read msb)
14  0 0 HI MOV B ad/lsb 0 MOV B       ( read lsb)
15  0 #B ad/csr MOV 04 # 1 MOV 0 SHR V ( start a/d ) .S
```

367 LIST

```
0 ( McCLELLAN SENSOR ELEMENT SAMPLING ROUTINE CONT'D 7/28/86 )
1 HEX
2   a/d# 1 MOV B 1 SHL  ( get offset for data table)
3   1 W MOV 0 ELEMENTS W) ADD  ( add to sum)
4   a/d# INC B 03 #B a/d# AND  ( inc pointer mod 4)
5   3 #B chan TEST 0=  ( if chan = 0, inc loop mod 4)
6   IF loopcnt INC B
7   3 #B loopcnt TEST 0=  ( if loop = 0, exit)
8   IF NEXT THEN
9   THEN chan INC B 3 #B chan AND  ( else inc chan )
10  AD/CONREG O MOV B chan 0 OR B  ( setup mux)
11  0 ad/mx/msb MOV B OF #B A/DSTATE MOV  ( save state)
12  L1 JMP
13 .S
14
15
```

368 LIST

```
0 ( McCLELLAN SENSOR DEVEL.          A/D SAMPLING      3/26/87)
1
2 HEX
3 CODE a/d.sample ( chan -- # read A/D chan )
4   AD/CONREG O MOV B                  ( load AD/CONREG in 0)
5   1 POP 1 0 OR B 0 ad/mx/msb MOV B  ( set mux to chan)
6   05 # 1 MOV BEGIN LOOP           ( delay 10 usec)
7   0 ad/csr MOV B                  ( start conv)
8   0 0 SUB OF #B A/DSTATE MOV     ( clear r0 & set flag)
9   wait                            ( wait for conversion)
10  0 #B A/DSTATE MOV              ( clear flag)
11  0 #B A/DFAILCTR MOV            ( clear count)
12  ad/mx/msb 0 MOV B              ( read a/d msb )
13  0 0 HI MOV B ad/lsb 0 MOV B    ( save & read lsb)
14  04 # 1 MOV 0 SHR V 0 PUSH     ( drop 4 lsb & add)
15  NEXT .S
```

369 LIST

```
0 ( McCLELLAN SENSOR DEVEL. AIR DENSITY SENSING 7/28/86)
1 HEX
2 CODE read_frequency ( Read Frequency Counter)
3   FREQFLAG 0 MOV B 0 0 OR B 0=      ( is FREQFLAG clear ?)
4   IF OE4 #B ckcsr MOV             ( latch status of ce1)
5   ce1 0 MOV B                   ( read status of ce1)
6   80 #B 0 TEST 0=              ( test for output low)
7   IF ODB #B ckcsr MOV           ( latch count of ce2)
8   FREQUENCY1 0 MOV 0 FREQUENCY MOV
9   ce2 0 MOV B ce2 0 HI MOV B ( read lsb, then msb )
10  0 NEG 0 FREQUENCY1 MOV        ( save results)
11  BO #B ckcsr MOV             ( write mode0/word to ce2)
12  0 0 SUB 0 ce2 MOV B 0 ce2 MOV B ( reset count)
13  1 #B FREQFLAG MOV           ( set flag)
14 THEN
15
```

370 LIST

```
0 ( McCLELLAN SENSOR DEVEL. AIR DENSITY SENSING 7/28/86)
1 HEX
2 ELSE OE4 #B ckcsr MOV          ( latch status of ce1)
3 ce1 0 MOV B                   ( read status of ce1)
4 80 #B 0 TEST 0= NOT           ( test for output high)
5 IF 0 #B FREQFLAG MOV          ( clear flag)
6 THEN
7 THEN NEXT .S
8
9
10
11
12
13
14
15
```

371 LIST

```
0 ( A/D SAMPLING CHANNEL 4 - 7                      3/14/89)
1
2 : A/D.READ ( READ CHANNEL 4 THRU 7 )
3   4 a/d.sample TEMPVOLT1 +! ( SUM TEMP. SEN. READING )
4   5 a/d.sample VBTEMP +!   ( SUM BARO. TEMP READINGS )
5   6 a/d.sample DUP VOLTS +! ( SUM 5 VOLT READING )
6   05VOLTS !
7   7 a/d.sample VBARD +! ;   ( SAVE 5 VOLT READING )
8
9 .S
10
11
12
13
14
15
```

372 LIST

```
0 ( McCLELLAN - SENSOR ELEMENT TEMPERATURE RANGE TEST 7/28/86)
1
2
3
4 .S
5
6
7
8
9
10
11
12
13
14
15
```

373 LIST

```
0 ( McCLELLAN SENSOR DATA VALIDATION ROUTINE 8/7/86)
1
2 : VALIDATE ( IF A/D READING EXCEEDS MAX VOLTAGE RANGE )
3     ( FOR THE PROCESSOR THEN THE LIKELY CAUSE )
4     ( IS AN A/D FAILURE )
5     256 05VOLTS @      ( FETCH LAST READING)
6     1433 3277 WITHIN   ( WITHIN 3.5-8 VOLT RANGE ?)
7     IF clear.status    ( CLEAR A/D FLAG )
8     ELSE set.status    ( SET A/D FLAG)
9     THEN ;
10    .S
11
12
13
14
15
```

374 LIST

```
0 ( McCLELLAN - DATA AQUISITION TASK 3/14/89)
1
2 : ACQUISITIONING ( -- ACQUIRE INPUT DATA & TEST FOR VALIDITY)
3 BEGIN PAUSE ACQUIRE C@
4     IF 0 ACQUIRE C!      ( CLEAR FLAG )
5     sample_elements       ( sample element voltages )
6     A/D.READ DACFLAG C@ ( sample other channels )
7     IF dac.load          ( load DAC if flag set )
8     THEN VALIDATE        ( test data range )
9     1 PROCESSFLAG C!     ( SET FLAG FOR PROCESS )
10    THEN
11    AGAIN ;
12 .S
13
14
15
```

378 LIST

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

379 LIST

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

380 LIST

0 HEX
1 CODE Q* (d d -- q)
2 83 C, EC C, 08 C, 8B C, FC C, 8B C, 45 C, 0E C, 8B C, 4D C,
3 0A C, F7 C, E1 C, 89 C, 55 C, 04 C, 89 C, 45 C, 06 C, 31 C,
4 C0 C, 89 C, 45 C, 02 C, 89 C, 05 C, 8B C, 45 C, 0C C, F7 C,
5 E1 C, 01 C, 45 C, 04 C, 11 C, 55 C, 02 C, A3 C, 15 C, 00 C,
6 8B C, 4D C, 08 C, 8B C, 45 C, 0E C, F7 C, E1 C, 01 C, 45 C,
7 04 C, 11 C, 55 C, 02 C, 83 C, 15 C, 00 C, 8B C, 45 C, 0C C,
8 F7 C, E1 C, 01 C, 45 C, 02 C, 11 C, 15 C, FA C, 83 C, C4 C,
9 10 C, FF C, 75 C, 06 C, FF C, 75 C, 04 C, FF C, 75 C, 02 C,
10 FF C, 35 C, FB C, NEXT
11
12 CODE Q+ (q q - q)
13 8B C, FC C, 8B C, 45 C, 06 C, 01 C, 45 C, 0E C, 8B C, 45 C,
14 04 C, 11 C, 45 C, 0C C, 8B C, 45 C, 02 C, 11 C, 45 C, 0A C,
15 8B C, 05 C, 11 C, 45 C, 08 C, 83 C, C4 C, 08 C, NEXT .S

381 LIST

```
0 ( 14-bit Fixed-point fraction arithmetic)
1 ( Snarfed from Poly-FORTH level 3 page 42 )
2
3 16384 CONSTANT +1
4
5 CODE *. ( n f - n) 0 POP 1 POP 1 IMUL
6 0 SHL 2 RCL 0 SHL 2 RCL 2 PUSH NEXT
7
8 CODE /. ( n n - f) 1 POP 2 POP 0 0 SUB
9 2 SAR 0 RCR 2 SAR 0 RCR 1 IDIV 0 PUSH NEXT
10
11 CODE U/ ( u u - u ) W POP 0 POP 2 2 SUB W DIV 0 PUSH NEXT
12
13 .S
14
15
```

382 LIST

```
0 ( Trigonometry for 14-bit fractions)
1 ( Snarfed from Poly-FORTH level 3 page 48 )
2
3 CODE TRIANGLE ( a - f) 0 POP 0 SHL 0 SHL
4 0\ IF 0 NEG THEN +1 # 0 SUB 0 NEG 0 PUSH NEXT
5
6 CODE 90- ( a - a) 0 POP 4096 # 0 SUB 0 PUSH NEXT
7
8 ( Hart 3300: 1.57079 -.64589 .07943 -.00433)
9 : COS ( a - f) TRIANGLE DUP DUP *. DUP -71 *.
10 1301 + OVER *. -10582 + *. 9352 + OVER *. + PAUSE ;
11
12 : SIN ( a - f) 90- COS ;
13 .S
14
15
```

383 LIST

```
0 ( Arc-tangent)
1 ( Snarfed from Poly-FORTH level 3 page 49 )
2
3 CODE !SET 0 POP 0 ROR CS NOT IF
4 LODS THEN R ) RCL NEXT
5
6 ( Hart 4960: .15920 -.05270 .02680 .41421)
7 HERE HEX -0C00 , 0400 , 3400 , -3C00 ,
8 1400 , -1C00 , -2C00 , 2400 , DECIMAL
9 : ATAN ( n n - a) DUP 0\ ) R ABS
10 SWAP DUP 0\ !SET NEGATE 2DUP ) !SET SWAP /.
11 DUP -6768 + SWAP 6786 *. +1 + /.
12 DUP DUP *. DUP 438 *. -864 + *. 2607 + *.
13 R) 2* LITERAL + @ + ABS PAUSE ;
14
15 .S
```

384 LIST

```

0 ( Angle conversion )
1 ( Shared from Poly-FORTH level 3 page 50 )
2
3 : DEG ( n - f ) 360 /. ;
4 : REV ( f - n ) 0 18000 8192 M*/ DROP ;
5 : (.5 ( f - f ) 8192 OVER < IF +1 - THEN ;
6
7
8
9
10 .S
11
12
13
14
15

```

385 LIST

```

0 ( DATA ACQUISITION HANDLER 9/APR/85 )
1
2 CODE SQRT ( d - n, RETURNS SQUARE ROOT FOR DOUBLE-LENGTH # )
3     2 POP 0 POP I PUSH W W SUB W I MOV 16 # 1 MOV
4     BEGIN 0 SHL 2 RCL I RCL 0 SHL 2 RCL I RCL
5         W SHL W SHL W INC W I CMP CS NOT IF
6         W I SUB W INC THEN W SHR
7     LOOP I POP W PUSH NEXT
8
9 .S
10
11
12
13
14
15

```

386 LIST

```

0 ( PROCESS QSQRT ROUTINE )
1
2 HEX CODE QSQRT ( q - d )
3 83 C, EC C, 04 C, 8B C, FC C, 31 C, D2 C, 31 C, C0 C, 89 C,
4 05 C, 89 C, 45 C, 02 C, B9 C, 20 C, 00 C, D1 C, 65 C, 0A C,
5 D1 C, 55 C, 08 C,
6 D1 C, 55 C, 06 C, D1 C, 55 C, 04 C, D1 C, D0 C, D1 C, D2 C,
7 D1 C, 65 C, 0A C, D1 C, 55 C, 08 C, D1 C, 55 C, 06 C, D1 C,
8 55 C, 04 C, D1 C, D0 C, D1 C, D2 C, D1 C, 25 C, D1 C, 55 C,
9 02 C, D1 C, 25 C, D1 C, 55 C, 02 C, 83 C, 05 C, 01 C, 83 C,
10 55 C, 02 C, 00 C, 3B C, 55 C, 02 C, 72 C, 12 C,
11 75 C, 04 C, 3B C, 05 C, 72 C, 0C C, 2B C, 05 C, 1B C, 55 C,
12 02 C, 83 C, 05 C, 01 C, 83 C, 55 C, 02 C, 00 C, D1 C, 6D C,
13 02 C, D1 C, 1D C, E2 C, B1 C, 8B C, 05 C, 8B C, 55 C, 02 C,
14 83 C, C4 C, 0C C, 50 C, 52 C, NEXT
15 .S

```

387 LIST

```
0 ( McCLELLAN - HEATER DUTY CYCLE ROUTINE      7/16/87 )
1 HEX
2 : HEAT.220 ( TURN HEATER OFF IF 110/220 FLAG IS CLEAR )
3     CONFIG C@ 20 AND      ( 110/220 FLAG RESET ? )
4         IF 80 clear.csrbit ( TURN OFF HEATER)
5     THEN ;
6
7 .S
8
9
10
11
12
13
14
15
```

388 LIST

```
0 ( McCLELLAN SENSOR HEATER ROUTINE 7/16/87)
1
2 : HEAT.CONTROL ( TEST TEMPERATURE & TURN HEATER ON ACCORDINGLY)
3     128 TEMPERATURE @      ( SET MASK & FETCH TEMP.)
4     HEATFLAG C@           ( USE EXPANDED RANGE ?)
5         IF -2200 1000 WITHIN ( WITHIN -22 - 10 DEG. C? )
6             IF set.csrbit   ( TURN HEATER ON )
7                 ELSE clear.csrbit ( TURN HEATER OFF )
8                     0 HEATFLAG C! ( CLEAR FLAG)
9             THEN
10            ELSE -2000 500 WITHIN ( WITHIN -20 - 5 DEG. C ? )
11                IF set.csrbit   ( TURN HEATER ON )
12                    1 HEATFLAG C! ( SET FLAG)
13                ELSE DROP      ( DROP 128 MASK)
14            THEN
15        THEN ; .S
```

389 LIST

```
0 ( McCLELLAN - VOLTAGE TO TEMPERATURE CONVERSION 7/28/86)
1
2 : A/D.ADJUST ( g,v,o -- av) ( {COUNT - OFFSET }*GAIN )
3     - M* 8192 0 D+ 14 dn/ ; ( GAIN IS SCALED TO 16384)
4
5 : D/A.ADJUST ( o,v,g -- av) ( [{2*COUNT * GAIN} + OFFSET]/SCALE)
6     M* 16384 0 D+ 13 dn/
7     + 2 n/ ; ( GAIN SCALED TO 16384)
8
9 : INTERP.TEMP ( volts -- degC LOOKUP & CONVERT VOLTS-TEMP)
10    BOTTOM @ 2* THERMIS @ + DUP >R ( SAVE ADDRESS)
11    @ - -1000 ( COMPUTE X - Xi )
12    R) 2@ ( FETCH Xi+1,Xi)
13    2DUP TPAIR 2! ( **** REMOVE)
14    - */ ( [Xi-Xi]*10/[Xi+1-Xi] )
15 ( ****) 8 BOTTOM @ - 1000 * + ; ( ADD DEGREE ) .S
```

390 LIST

```
0 ( McCLELLAN SENSOR VOLTAGE TO TEMPERATURE ROUTINE 3/14/89)
1
2 : G&O.ADJUST ( n -- n ADJUST FOR NON-IDEAL AFFECTS)
3     GAINS/OFFSETS @ 16 + 2@ ( RETRIEVE GAIN & OFFSET)
4     ROT SWAP A/D.ADJUST ; ( CORRECT TEMP READING)
5
6 : V/T.ADJUST ( -- USE DATA IN GAINS/OFFSET TABLE TO ADJUST)
7     TEMPVOLT1 @ G&O.ADJUST 2+ 4 / ( CORRECT TEMP READING)
8     VOLTS @ G&O.ADJUST 2+ 4 / ( CORRECT VOLTAGE READING)
9     TEMPVOLT1 2@ HOLDBUF 12 + 2! ( **** TEMPORARY TEST USE)
10    DUP CSVOLTS !
11    O TEMPVOLT1 ! O VOLTS ! ; ( SAVE CORRECTED 5VOLTS)
12
13 : OVER.HEAT ( n -- n RETURN DESIRED ELEMENT TEMPERATURE)
14     OVERHEAT @ + ; ( COMPUTE OVERHEAT)
15 .S
```

391 LIST

```
0 ( McCLELLAN SENSOR VOLTAGE TO TEMPERATURE ROUTINE 7/31/86)
1
2 : CONVERT.V/T ( -- )
3     SENSORSTAT @ 256 AND NOT ( NO A/D FAILUES ?)
4     IF 1024 V/T.ADJUST ( ADJUST FOR A/D GAIN & OFFSET)
5     SVOLTS SWAP */ ( TVOLTS*VOLTAGE/SVOLTS=ADJ/READING)
6     THERMIS @ ( FETCH POINTER TO TABLE )
7     SEARCH.SETUP ( SETUP FOR SEARCH)
8     BSEARCH ( FIND  $X_i$  &  $X_{i+1}$  FOR INTERPOLATION)
9     INTERP.TEMP ( FIND  $X_i$  &  $X_{i+1}$  & INTERPOLATE)
10    DUP -5500 7600 WITHIN ( WITHIN RANGE ?)
11    IF TEMPERATURE ! ( YES, SAVE RESULTS)
12    clear.status
13    ELSE DROP set.status ( NO, DROP RESULTS & SET FLAG)
14    THEN
15    THEN ; .S
```

392 LIST

```
0 ( McCLELLAN SENSOR ELEMENT RESISTANCE & D/A OUTPUT 10/12/86)
1
2 : EL.TAB ( n -- a FIND ADDRESS TO 1 OF 4 ELEMENT TABLES)
3     26 * ELEMENTOHMS @ + ;
4
5 : INTERP.EXTRAP? ( n -- n, yi, r, q n=DESIRED ELEMENT TEMP )
6     DUP DUP 5000 17000 WITHIN ( INTERPOLATE ?)
7     IF 1000 /MOD ( r=X-Xi, q=index1)
8     5 - 0 SWAP ( COMPUTE INDEX, MAKE r A d)
9     ELSE DUP 5000 < ( EXTRAPOLATE DOWN ?)
10    IF DUP 0<
11        IF 0 1 - ELSE 0 ( MAKE DOUBLE, POS. OR NEG. )
12        THEN 5000 0 D- 0 ( SET [X-Xi] AS d & INDEX)
13        ELSE 16000 - 0 11 ( SET TO EXTRAPOLATE UP)
14        THEN
15    THEN ; .S
```

393 LIST

```
0 ( McCLELLAN SENSOR ELEMENT RESISTANCE & D/A OUTPUT 10/12/86)
1
2 : INTERP.ELRES ( n, i -- n FIND ELEMENT RESISTANCE & D/A OUTPUT)
3     DUP >R EL.TAB >R      ( SAVE INDEX & TABLE ADDR.)
4     INTERP.EXTRAP?        ( SET UP TO ESTIMATE RESISTANCE)
5     2* R> + 2@ 2DUP RPAIR 2! ( **** ( FETCH Yi+1, Yi )
6     DUP >R - 1000 M*/      ( SAVE Yi, COMPUTE [X-Xi][Yi+1-Yi])
7     R) 0 D+ 50000 0 D+    ( S=RESISTANCE IN 10 uOHMS)
8     2DUP RESISTANCE I 4 * + 2! ( SAVE FOR POWER CALCULATION)
9     4096 8000 M*/
10    3 0 D+ 5 M/          ( COMPUTE IDEAL D/A OUTPUT)
11    I 4 * GAINS/OFFSETS @ + 2@ ROT SWAP ( SETUP FOR ADJUST)
12    D/A.ADJUST           ( ADJUST FOR NON-IDEAL D/A)
13    R> 2* D/AOUTBUF + ! ; ( SAVE IN D/A ARRAY) .S
14 .S
15
```

394 LIST

```
0 ( McCLELLAN SENSOR ELEMENT RESISTANCE ROUTINE 7/31/86)
1
2 : ELEMENT.RESISTANCE ( -- )
3     TEMPERATURE @        ( TEST FOR VALID TEMP)
4     OVER.HEAT 4 0         ( COMPUTE NEW OVERHEAT)
5     DO I INTERP.ELRES ( FIND ELEMENT RESISTANCE & D/A VALUE)
6     PAUSE
7     LOOP OVERHEATSETPT ! ;
8
9 : NEW.TEMP ( -- da4,da3,da2,da1 )
10    D/AOUTBUF DUP >R
11    6 + @ I 4 + @ I 2+ @ R> @ ; ( PUT DATA ON STACK)
12
13 .S
14
15
```

395 LIST

```
0 ( McCLELLAN - SENSOR ELEMENT TEMPERATURE SET FUNCTION 7/28/86)
1
2 : SET_ELEMENT_TEMP ( -- READ TEMP, TEST & OUTPUT NEW TEMP)
3     CONVERT.V/T          ( CONVERT VOLTAGE TO TEMP)
4     ELEMENT.RESISTANCE   ( DETERMINE ELEMENT RESISTANCE)
5     NEW.TEMP d/astuff    ( DETERMINE NEW TEMP SETTING & OUTPUT)
6     HEAT.CONTROL ;       ( 25% DUTY CYCLE FOR 220 VOLT AC)
7
8 .S
9
10
11
12
13
14
15
```

396 LIST

```
0 ( McCLELLAN SENSOR PRESSURE COMPENSATION ROUTINE 3/14/89)
1
2 : BTEMP.LOOKUP ( -- btemp LOOKUP BTEMP. FOR INDEXING)
3     BTEMPBAR @ BTEMPDEX @          ( GET btemp & TABLE)
4     SEARCH.SETUP BSEARCH          ( SEARCH TABLE)
5     BOTTOM @ DUP BTBOTTOM !      ( SAVE INDEX)
6     2* BTEMPDEX @ + 2@ BTPAIR 2! ; ( SAVE btemp PAIR )
7
8 : BARO.LOOKUP ( btemp -- btemp, baro LOOKUP PRESSURE FOR INDEXG)
9     BAROBAR @ BARODEX @          ( GET baro & TABLE)
10    SEARCH.SETUP BSEARCH          ( SEARCH TABLE)
11    BOTTOM @ DUP BBOTTOM !      ( SAVE INDEX)
12    2* BARODEX @ + 2@ BPAIR 2! ; ( SAVE barod PAIR )
13
14
15 .S
```

397 LIST

```
0 ( McCLELLAN SENSOR PRESSURE COMPENSATION ROUTINE 3/14/89)
1
2 : BCOMP.LOOKUP ( btemp, baro, cta -- btemp, baro FIND PAIRS )
3     BBOTTOM @ 1-              ( COMPUTE # OF ROWS TO 1st PAIR )
4     BTEMPDEX @ @             ( START OF TABLE HAS LENGTH )
5     DUP >R *
6     BTBOTTOM @ 1- 2* +       ( ADD INDEX TO POINT TO COLUMN)
7     + DUP             ( ADD mca TO POINT TO 1st PAIR OF INTEREST)
8     2@ LBCPAIR 2!           ( SAVE LOWER COMPENSATION PAIR)
9     R) +
10    2@ UBCPAIR 2! ;         ( INCREMENT TO THE NEXT ROW)
11    2@ UBCPAIR 2! ;         ( SAVE UPPER COMPENSATION PAIR )
12
13
14
15 .S
```

398 LIST

```
0 ( McCLELLAN SENSOR PRESSURE COMPENSATION ROUTINE 3/14/89)
1
2 : BARO.INTERP ( btemp, baro -- comp )
3     )R LBCPAIR @ SWAP ( SAVE S, GET BC1A1, btemp FIRST )
4     BTPAIR @ - DUP >R ( COMPUTE A-A1, SAVE COPY)
5     LBCPAIR 2@ -        ( COMPUTE MC1Au-MC1A1)
6     BTPAIR 2@ - DUP >R ( COMPUTE Au-A1, SAVE COPY)
7     */ + DUP           ( FINAL ANSWER MC1)
8     UBCPAIR DUP @      ( GET MCuA1)
9     SWAP 2@ -           ( COMPUTE MCuAu-MCuA1)
10    2R) */ +
11    SWAP -              ( RETRV A-A1 & Au-A1 COMP. MCu)
12    R) BPAIR @ -        ( COMPUTE S - S1 )
13    BPAIR 2@ -           ( COMPUTE Su-S1 )
14    */ + ;              ( FINAL ANSWER, MC)
15 .S
```

399 LIST

```
0 ( McCLELLAN SENSOR VOLTAGE TO PRESSURE ROUTINE 3/14/89)
1
2 : B/T.ADJUST ( -- v1, v2 USE GAINS/OFFSET TABLE TO ADJUST )
3     VBTEMP @ G&O.ADJUST      ( CORRECT TEMP READING)
4     VBARO @ G&O.ADJUST      ( CORRECT PRESSURE READING)
5     0 VBTEMP !   0 VBARO ! ; ( CLEAR VARIABLES)
6
7 : +M! ( n, a -- ADD n TO CONTENTS OF DOUBLE AT a )
8     DUP >R 2@ ROT M+ R> 2! ;
9
10
11 .S
12
13
14
15
```

400 LIST

```
0 ( McCLELLAN SENSOR PRESSURE TRANSDUCER HANDLER ROUTINE 3/14/89)
1
2 : BP.SUMS ( -- PERFORM SUMMING OF THE PRESSURE INPUTS )
3     B/T.ADJUST      ( CORRECT FOR OFFSET & GAIN )
4     SVOLTS CSVOLTS @ */ ( NORMALIZE TO 5 VOLTS )
5     BAROSUM +M!      ( SUM TO GET AVERAGE )
6     SVOLTS CSVOLTS @ */ ( NORMALIZE TO 5 VOLTS )
7     BTEMPSUM +M! ;
8 .S
9
10
11
12
13
14
15
```

401 LIST

```
0 ( McCLELLAN SENSOR VOLTAGE TO PRESSURE ROUTINE 3/14/89)
1
2 : AIR.PRESSURE ( COMPENSATE PRESSURE & SAVE )
3     8192          ( PLACE FLAG ON ST. )
4     BTEMP.LOOKUP    ( FIND TEMP. INDEX )
5     BARO.LOOKUP    ( FIND PRESSURE INDEX )
6     BAROCOMP @ BCOMP.LOOKUP ( FIND COMP. PAIRS )
7     BARO.INTERP    ( COMPENSATE PRESSURE )
8     DUP 5500 10500 WITHIN ( GOOD RANGE ? )
9     IF PRESSURE ! clear.status ( SAVE RESULTS )
10    ELSE DROP set.status ( NO, SO SET BIT )
11    THEN ;
12 .S
13
14
15
```

402 LIST

```
0 ( McCLELLAN SENSOR AIR PRESSURE DETERMINATION ROUTINE 3/14/89)
1
2 : COMPUTE.PRESSURE ( -- READ FREQUENCY EVERY 10 SECONDS )
3     SECCNT C@ ?DUP      ( COUNTER = 0 ? )
4     IF 1-                ( NO, SO DECREMENT IT )
5     ELSE 7                ( RESET COUNTER, 8 SECS )
6     BAROSUM 2@ 2DUP BAROTOT 2! ( ***** REMOVE )
7           4 M+ 3 dn/ ( FIND AVERAGE OF 8 READINGS )
8     BAROBAR !            ( SAVE IT )
9     BTEMPSUM 2@ 2DUP BTEMPTOT 2! ( ***** REMOVE )
10           4 M+ 3 dn/ ( SAME )
11     BTEMPBAR !
12     O O BAROSUM 2!      ( CLEAR SUMS )
13     O O BTEMPSUM 2!
14     AIR.PRESSURE        ( COMPENSATE PRESSURE )
15     THEN SECCNT C! ;    ( SAVE SECONDS COUNT )
```

403 LIST

```
0 ( McCLELLAN SENSOR PRESSURE TRANSDUCER HANDLER ROUTINE 3/14/89)
1
2 : PT.HANDLER ( IF ALL TESTS ARE OK, CONVERT FREQ TO PRESSURE )
3     SENSORSTAT @        ( FETCH STATUS TO TEST )
4     256 AND NOT         ( NO A/D FAILURES ? )
5     PHASE C@ PHASE? C! ( ***** REMOVE )
6     IF PTINIT C@        ( 1st PASS ? )
7     IF BP.SUMS          ( CORRECT FOR OFFSET & GAINS )
8     ELSE 240 PTINIT C! ( SET FLAG )
9           O VBTEMP ! O VBARO ! ( CLEAR SAMPLES )
10    THEN
11    COMPUTE.PRESSURE ( DETERMINE AIR PRESSURE )
12    THEN ; .S
13
14
15
```

404 LIST

```
0 ( McCLELLAN SENSOR VELOCITY COMPENSATION ROUTINE 6/20/87)
1
2 : ANGLE.LOOKUP ( -- theta LOOKUP ANGLE FOR INDEXING )
3     DIRECTION @ ANGLEDEX @ ( GET theta & TABLE )
4     SEARCH.SETUP BSEARCH   ( SEARCH TABLE )
5     BOTTOM @ DUP ABOTTOM ! ( SAVE INDEX )
6     2* ANGLEDEX @ + 2@ APAIR 2! ; ( SAVE theta PAIR )
7
8 : MAG.LOOKUP ( theta -- theta, speed LOOKUP SPEED FOR INDEXING )
9     SPEED @ MAGDEX @      ( GET speed & TABLE )
10    SEARCH.SETUP BSEARCH   ( SEARCH TABLE )
11    BOTTOM @ DUP MBOTTOM ! ( SAVE INDEX )
12    2* MAGDEX @ + 2@ MPAIR 2! ; ( SAVE speed PAIR )
13
14
15 .S
```

405 LIST

```
0 ( McCLELLAN SENSOR VELOCITY COMPENSATION ROUTINE 6/20/87)
1
2 : COMP.LOOKUP ( theta, speed, cta -- theta, speed FIND PAIRS )
3     MBOTTOM @ 1-          ( COMPUTE # OF ROWS TO 1st PAIR )
4     ANGLEDEX @ @         ( START OF TABLE HAS LENGTH )
5     DUP >R *
6     ABOTTOM @ 1- 2* +    ( ADD INDEX TO POINT TO COLUMN)
7     + DUP               ( ADD mca TO POINT TO 1st PAIR OF INTEREST)
8     2@ LMCPAIR 2!        ( SAVE LOWER COMPENSATION PAIR)
9     R) +
10    2@ UMCPAIR 2! ;     ( INCREMENT TO THE NEXT ROW)
11                               ( SAVE UPPER COMPENSATION PAIR)
12 .S
13
14
15
```

406 LIST

```
0 ( McCLELLAN SENSOR VELOCITY COMPENSATION ROUTINE 6/20/87)
1
2 : INTERP.INTERP ( theta, speed -- comp )
3     >R LMCPAIR @ SWAP ( SAVE S, GET MC1A1, theta 1st)
4     APAIR @ - DUP >R   ( COMPUTE A-A1, SAVE COPY)
5     LMCPAIR 2@ -       ( COMPUTE MC1Au-MC1A1)
6     APAIR 2@ - DUP >R   ( COMPUTE Au-A1, SAVE COPY)
7     */ + DUP           ( FINAL ANSWER MC1)
8     UMCPAIR DUP @      ( GET MCuA1)
9     SWAP 2@ -           ( COMPUTE MCuAu-MCuA1)
10    2R) */ +
11    SWAP -
12    R) MPAIR @ -        ( COMPUTE S - S1 )
13    MPAIR 2@ -          ( COMPUTE Su-S1 )
14    */ + ;              ( FINAL ANSWER, MC)
15 .S
```

407 LIST

```
0 ( McCLELLAN SENSOR VELOCITY COMPENSATION ROUTINE 7/31/86)
1
2 : UPDATE.SSECTAIL ( INCREMENT TAIL POINTER MODULO 5 )
3     SSECTAIL DUP C@ 1+ 5 MOD .
4     SWAP C! ;
5
6 : SSECQ.STORE ( ycomp, ycomp -- STORE COMPENSATED VALUES )
7     2DUP XCOMP 2!        ( CLEAR STACK FROM POLAR.REC ****)
8     X5SECQ                ( PLACE QUE ADDRESS ON STACK)
9     SSECTAIL C@ 4 *      ( COMPUTE OFFSET TO TAIL)
10    DUP >R + 2!          ( SAVE & ADD TO ADDRESS & STORE)
11    2DUP YCOMP 2!
12    Y5SECQ R) + 2! UPDATE.SSECTAIL ;
13 .S
14
15
```

408 LIST

```
0 ( VECTORIAL AVERAGE ROUTINE 3/4/86)
1
2 : DVEC.AVE ( 2x, 2y -- Zresultant, R=[x*x + y*y]SQRT )
3     2)R DABS 2DUP Q*
4     2R) DABS 2DUP Q* Q+ SQRT PAUSE ; ( COMPUTE DOUBLE ROOT)
5
6 : VEC.AVE ( x, y -- resultant, R=[x*x + y*y]SQRT )
7     DUP M* ROT DUP M* D+ SQRT ;
8
9
10 .S
11
12
13
14
15
```

409 LIST

```
0 ( POLAR TO RECTANGULAR CONVERSION 9/26/86)
1
2 : REC.POLAR ( 2x, 2y -- )
3     2DUP 2)R 2SWAP 2DUP 2)R      ( SAVE X & Y )
4     DVEC.AVE 2 O D+ 4 M/        ( COMPUTE MAGNITUDE)
5     2R) 4 M/ 2R) 4 M/ SWAP ATAN ( COMPUTE DIRECTION)
6     DIRECTION ! SPEED ! ;     ( SAVE RESULTS )
7
8 : POLAR.REC ( m, a -- 2y, 2x )
9     2DUP COS M* 1 T* 2048 T/ 2)R      ( COMPUTE X )
10    SIN M* 1 T* 2048 T/ 2R) ;       ( COMPUTE Y )
11
12 .S
13
14
15
```

410 LIST

```
0 ( McCLELLAN SENSOR VELOCITY COMPENSATION ROUTINE 7/31/86)
1
2 : INTERP.TC ( n -- n  INTERPOLATE TEMPERATURE COMPENSATION)
3     BOTTOM @ 2* TEMPTAB + 2@ ( FETCH PAIR FROM TABLE)
4     2)R I' - -10          ( SAVE PAIR, FIND X-Xi )
5     2R) - */
6     BOTTOM @ -10 *        ( COMPUTE MINOR OFFSET)
7     1030 + + ;           ( ADD Y BASE & QUOTIENT)
8
9 .S
10
11
12
13
14
15
```

411 LIST

```
0 ( McCLELLAN SENSOR VELOCITY COMPENSATION ROUTINE 7/31/86)
1
2 : TEMP.COMP ( ctheta, cspeed -- cspeed, ctheta TEMP. COMP.)
3     TEMPERATURE @          ( FETCH TEMP. )
4     TEMPTAB SEARCH.SETUP   ( SETUP FOR TABLE SEARCH )
5     BSEARCH INTERP.TC      ( INTERPOLATE CORRECTION)
6     DUP TEMPCO !          ( ***** REMOVE )
7     1000 SWAP */          ( COMPENSATE THE MAGNITUDE )
8     DUP SPEEDLIMIT @ >    ( TEST FOR OUT OF RANGE)
9     IF 4096 set.status     ( YES, SET FLAG)
10    ELSE 4096 clear.status ( NO, CLEAR STATUS)
11    THEN SWAP ;           ( SETUP FOR POLAR.REC )
12 .S
13
14
15
```

412 LIST

```
0 ( McCLELLAN SENSOR VELOCITY COMPENSATION ROUTINE 7/31/86)
1
2 : DENSITY.COMP ( ctheta -- ctheta, cspeed DENSITY COMP. MAG. )
3     DUP CTHETA !          ( ***** TEST SUPPORT )
4     SPEED @ MCOMP @ -
5     PRESREF @ PRESSURE @ */ ( DENSITY COMPENSATE )
6     ;
7 .S
8
9
10
11
12
13
14
15
```

413 LIST

```
0 ( McCLELLAN SENSOR VELOCITY COMPENSATION ROUTINE 7/31/86)
1
2 : SSUM LOOP ( 11, 12 -- d  PRODUCE SUM OF 5 1 SEC VALUES)
3     DO I 2@ D+ 4
4     +LOOP ;
5
6 : SCALE.DOWN ( d -- n  ROUNDED SCALING )
7     DUP 0( IF -20 -1 ELSE 20 0 THEN D+ 40 M/ ;
8
9 .S
10
11
12
13
14
15
```

414 LIST

```
0 ( McCLELLAN SENSOR VELOCITY COMPENSATION ROUTINE 7/31/86)
1
2 : SSEC.AVE ( PRODUCE SUM OF 5 1 SEC X & Y VALUES )
3     0 0          ( SETUP STACK)
4     X5SECQ 20 SET ( SET LOOP LIMITS )
5     SSUM.LOOP      ( SUM THE QUE VALUES )
6     SCALE.DOWN
7     XBAR ! 0 0      ( STORE X COORDINATE )
8     YSSECQ 20 SET ( DO THE SAME FOR Y )
9     SSUM.LOOP
10    SCALE.DOWN
11    YBAR ! ;       ( STORE Y COORDINATE )
12 .S
13
14
15
```

415 LIST

```
0 ( McCLELLAN SENSOR ELEMENT POWER      ROUTINE      7/25/87)
1
2 : MAG.COMP ( MAGNITUDE COMPENSATION )
3     ANGLE.LOOKUP MAG.LOOKUP ( SET TABLE INDICES)
4     MAGCOMP @ COMP.LOOKUP ( LOOKUP COMP. VALUES)
5     INTERP.INTERP MCOMP ! ; ( INTERPOLATE & COMPENSATE)
6
7 : ANGLE.COMP ( -- ctheta ANGLE COMPENSATE theta )
8     ANGLECOMP @ COMP.LOOKUP ( LOOKUP COMP. VALUES )
9     DIRECTION @ DUP SPEED @ ( SETUP STACK)
10    INTERP.INTERP - DUP 0\ ( INTERP., 0 POLE WRAP ? )
11    IF 16384 +
12    ELSE DUP 16384 )      ( YES, WRAP IT TO 360 )
13    IF 16384 -
14    THEN
15    THEN ; .S
```

416 LIST

```
0 ( McCLELLAN SENSOR ELEMENT POWER      ROUTINE      2/25/87)
1
2 : SUM.LOOP ( 11, 12 -- d PRODUCE 1 SEC UNCOMPENSATED VELOSITIES)
3     DO I @ M+ 2      ( FETCH VALUES & ADD)
4     +LOOP ;
5
6 : SUM.QUARTERS ( SUM RAW X & Y VALUES )
7     0 0 XRAWQ 8 SET SUM.LOOP ( SUM X VALUES )
8     XUNCOMP 2!          ( SAVE RESULT)
9     0 0 YRAWQ 8 SET SUM.LOOP ( SUM Y VALUES )
10    YUNCOMP 2! ;        ( SAVE RESULT)
11
12 .S
13
14
15
```

417 LIST

```
0 ( McCLELLAN SENSOR VELOCITY COMPENSATION ROUTINE 7/31/86)
1
2 : VELOCITY_COMP ( -- PRODUCE COMPENSATED VELOSITIES)
3     no.op2 ( *****REMOVE *****)
4         1 DACFLAG C! SUM.QUARTERS ( FIND 1 SEC VALUES)
5         XUNCOMP 2@ YUNCOMP 2@ REC.POLAR
6         MAG.COMP PAUSE      ( COMPENSATE MAGNITUDE )
7         ANGLE.COMP PAUSE   ( COMPENSATE ANGLE)
8         DENSITY.COMP      ( COMPENSATE FOR DENSITY)
9         TEMP.COMP POLAR.REC ( COMPENSATE FOR TEMP. )
10        5SECQ.STORE 5SEC.AVE ( PRODUCE 5 SEC AVERAGE)
11        HEAT.220          ( DUTY CYCLE THE HEATER )
12        PAUSE PT.HANDLER   ( PRESSURE SENSOR HANDLER )
13        no.op3 ( ***** REMOVE *****)
14                           ; ( PRODUCE 5 SEC AVERAGE )
15 .S
```

418 LIST

```
0 ( McCLELLAN SENSOR ELEMENT POWER      ROUTINE      2/25/87)
1
2 CODE UM/ ( d, u -- q PERFORMS UNSIGNED DIVISION & RETURNS SING)
3     W POP 2 POP 0 POP W DIV 0 PUSH NEXT
4
5 CODE D2/RND ( d -- d/2 DIVIDES DOUBLE BY 2 RETURN ROUNDED SING)
6     1 POP 0 POP 1 # 0 ADD 0 # 1 ADC
7     1 SAR 0 RCR 0 PUSH NEXT
8
9 : CLR.RESET ( CK IF RESET TIMER IS 0 TO CLEAR ANY RESET ERROR )
10    RESETIMER @ ?DUP           ( TIMER STILL RUNNING ?)
11    IF 1- DUP RESETIMER ! 0=   ( DECREMENT TIMER, = 0? )
12    IF 0 RESETCTR C!
13        16 clear.status       ( CLR # OF STARTUPS )
14    THEN
15    THEN ;                   .S
```

419 LIST

```
0 ( McCLELLAN SENSOR ELEMENT POWER      ROUTINE      2/25/87)
1
2 : ELEMENT.POWER ( -- COMPUTE ELEMENT POWER )
3     4 0                      ( DO ALL FOUR ELEMENTS )
4     DO ELEMENTS I 2* + @      ( RETURN DIGITIZED VOLTAGE)
5     G&O.ADJUST               ( ADJUST FOR A/D GAIN & O.S. )
6     2* DUP M*
7     RESISTANCE I 4 * + 2@    ( RETRIEVE RESISTANCE)
8     D2/RND                   ( DIVIDE BY 2)
9     DUP >R 0 D2/RND M+      ( SAVE DIVISOR, PRE-ROUND)
10    R> UM/ ELEMENTPOWER
11    I 2* + !
12    LOOP ;
13 .S
14
15
```

420 LIST

```

0 ( McCLELLAN SENSOR ELEMENT POWER DIFFERENCE ROUTINE 2/25/87)
1
2 : POWER.DIFF ( COMPUTE POWER DIFFERENCE BETWEEN ELEMENT PAIRS)
3     ELEMENTPOWER DUP           ( PLACE ADDRESS ON STACK)
4     2@ SWAP - XPOWER !       ( COMPUTE X+ - X- )
5     4 + 2@ SWAP - YPOWER ! ; ( COMPUTE Y+ - Y- )
6
7 : POWER.SUM ( a -- xp COMPUTE POWER SUMS OF ELEMENT PAIRS)
8     2@ 2DUP             ( PLACE AXIS PAIR ON STACK)
9     + ROT ROT <         ( COMPUTE SUM, X+ < X- ? )
10    IF NEGATE          ( CORRECT SIGN )
11    THEN ;
12
13 : !.RAW ( n, a -- STORE n IN QUE AT a )
14     RAWTAIL C@ 2* + ! ;
15 .S

```

421 LIST

```

0 ( McCLELLAN SENSOR RAW VELOCITY ROUTINE 7/31/86)
1
2 : INTERP.RAW ( n, a -- PULL X & Y PAIRS FORM TABLE @ a & INTERP)
3     DUP >R             ( SAVE ADDRESS)
4     BOTTOM @ 2* +        ( CALCULATE table[bottom])
5     DUP >R @ -          ( SAVE ADDRESS, COMPUTE X-Xi )
6     I 2@ -              ( FETCH & COMPUTE Xi+1 - Xi )
7     R) R) @ + 4 +        ( COMPUTE ADDRESS OF CO-TABLE)
8     DUP >R 2@ -          ( FETCH & COMPUTE Yi+1 - Yi)
9     SWAP */              ( IY = [X-Xi][Yi+1-Yi]/[Xi+1-Xi] )
10    R) @ + ;            ( Y = Yi + IY )
11
12 : X.RAW ( px -- vxraw INTERPOLATE VELOCITY FROM X POWERS )
13     PDX @ SEARCH.SETUP   ( SET UP FOR BINARY SEARCH)
14     BSEARCH               ( SEARCH PDX )
15     PDX @ INTERP.RAW ;    ( INTERPOLATE) .S

```

422 LIST

```

0 ( McCLELLAN SENSOR RAW VELOCITY ROUTINE 7/31/86)
1
2 : Y.RAW ( pdy -- vxraw INTERPOLATE VELOCITY FROM Y POWERS )
3     PDY @ SEARCH.SETUP   ( SET UP FOR BINARY SEARCH)
4     BSEARCH PDY @ INTERP.RAW ; ( SEARCH & INTERPOLATE)
5
6 : UPDATE.RAWTAIL ( INCREMENT TAIL POINTER MOD 4 )
7     RAWTAIL DUP C@ 1+ 3 AND SWAP C! ;
8
9 : RAIN.? ( vraw, a -- vraw, a CHECK FOR RAIN SPIKES )
10    DUP >R RAWTAIL C@ 2* + @      ( GET LAST vraw )
11    OVER - ABS RAINLIMIT )        ( CURRENT - LAST )
12    IF 2/ 0 0 I 8                ( IF OVER LIMIT )
13    SET SUM LOOP ROT M+ 5 M/    ( AVERAGE ALL )
14    THEN R) ; .S
15

```

423 LIST

```
0 ( SENSOR SYSTEM ELEMENT DRIVER SELF-TEST ROUTINE      9/03/87 )
1
2 : TEST.ELEMENTS ( ELEMENT DRIVER TEST )
3     512 4 0          ( SET MASK VALUE & LOOP LIMITS)
4     DO ELEMENTS I 2* + @ ( CHECK VOLTAGE ON EACH )
5         1000 16000 WITHIN
6         LOOP AND AND AND ( AND RESULTS )
7             IF clear.status ( CLEAR STATUS IF OK )
8                 0 ELECTR C! ( CLEAR COUNT )
9             ELSE ELECTR DUP C@ ( CK COUNT )
10            1+ 18 MIN DUP ROT C! 18 =
11            IF set.status ( SET STATUS IF FAILED)
12            ELSE DROP
13            THEN
14            THEN ;
```

424 LIST

```
0 ( McCLELLAN SENSOR  RAW VELOCITY ROUTINE    7/31/86)
1
2 : SUM.YPOWERS? ( yyraw -- yyraw   POWER SUM SCALING)
3     DUP ABS YLOLIMIT @ >          ( ABOVE LO LIMIT ? )
4     IF YSFLAG C@                  ( SCALING FLAG SET ? )
5     IF DUP ABS YHILIMIT @ >      ( ABOVE HI LIMIT ? )
6     IF DROP ELEMENTPOWER 2+ POWER.SUM ( SUM YPOWERS)
7         Y.RAW YSCALE @ 16384 */ ( SCALE VELOCITY )
8         THEN
9     ELSE DUP ELEMENTPOWER 2+ POWER.SUM ( SUM YPOWER )
10    Y.RAW 16384 SWAP */ YSCALE ! ( COMPUTE SCALE)
11    1 YSFLAG C!                  ( SET SCALE FLG)
12    THEN
13    ELSE 0 YSFLAG C!           ( BELOW LIMIT, CLR FLG)
14    THEN ; .S
```

425 LIST

```
0 ( McCLELLAN SENSOR  RAW VELOCITY ROUTINE    7/31/86)
1
2 : SUM.XPOWERS? ( xvraw -- xvraw   POWER SUM SCALING)
3     DUP ABS XLOLIMIT @ >          ( ABOVE LO LIMIT ? )
4     IF XSFLAG C@                  ( SCALING FLAG SET ? )
5     IF DUP ABS XHILIMIT @ >      ( ABOVE HI LIMIT ? )
6     IF DROP ELEMENTPOWER POWER.SUM ( SUM XPOWERS)
7         X.RAW XSCALE @ 16384 */ ( SCALE VELOCITY )
8         THEN
9     ELSE DUP ELEMENTPOWER POWER.SUM ( SUM XPOWERS )
10    X.RAW 16384 SWAP */ XSCALE ! ( COMPUTE SCALE )
11    1 XSFLAG C!                  ( SET SCALE FLAG)
12    THEN
13    ELSE 0 XSFLAG C!           ( BELOW LIMIT, CLR FLG)
14    THEN ; .S
```

426 LIST

```
0 ( McCLELLAN SENSOR  RAW VELOCITY ROUTINE    7/31/86)
1
2 : RAW_VELOCITY ( COMPUTE RAW VELCITIES EVERY QUARTER SECOND)
3     TEST_ELEMENTS          ( CHECK DRIVERS)
4     ELEMENT_POWER POWER_DIFF ( COMPUTE POWERS )
5     XPOWER @ X_RAW          ( SUM_XPOWERS? ( COMPUTE X RAWS )
6     XRAWQ RAIN.? !.RAW      ( STORE RESULTS)
7     YPOWER @ Y_RAW          ( SUM_YPOWERS? ( COMPUTE Y RAWS )
8     YRAWQ RAIN.? !.RAW      ( STORE RESULTS)
9     ELEMENTS HOLDBUF 36 CMOVE ( UPDATE. RAWTAIL ( UPDATE QUES)
10    1 PRINTFLAG C! ;        ( MOVE DATA VALUES)
11
12 .S
13
14
15
```

427 LIST

```
0 ( McCLELLAN SENSOR  RAW VELOCITY ROUTINE    3/14/89)
1
2 : 1ST_RAW ( COMPUTE FIRST RAW VELCITIES)
3     ELEMENT_POWER POWER_DIFF ( COMPUTE POWERS )
4     XPOWER @ X_RAW          ( COMPUTE X RAWS )
5     XRAWQ !.RAW            ( STORE RESULTS)
6     YPOWER @ Y_RAW          ( COMPUTE Y RAWS )
7     YRAWQ !.RAW ;          ( STORE RESULTS )
8
9
10
11
12
13
14
15
```

428 LIST

```
0 ( McCLELLAN SENSOR  DATA PROCESSING TASK           7/28/86)
1
2 : PROCESSING ( DETERMINE STATE OF 8 PHASE 1/4 SECOND COUNTER )
3     BEGIN PAUSE PROCESSFLAG C@ 
4     UNTIL 0 PROCESSFLAG C! 1ST_RAW   ( INITIALIZE RAWQS )
5     BEGIN PROCESSFLAG C@          ( START PROCESSING ? )
6     IF 0 PROCESSFLAG C!
7         RAW_VELOCITY PAUSE       ( PROCESS RAW SPEED )
8     PHASE C@ DUP PHASE? C! 3 AND DUP 0= ( PHASE COUNT = 0 OR 4 ?)
9         IF DROP VELOCITY_COMP PAUSE ( DETER TRUE VELOCITY)
10        ELSE 3 =
11            IF SET_ELEMENT_TEMP PAUSE ( PHASE = 3 OR 7 ?)
12            THEN
13            THEN CLR_RESET          ( CK IF TIME TO CLR RESET)
14            THEN PAUSE
15            AGAIN ;               .S
```

432 LIST

```

0 ( McCLELLAN BUILT IN TEST STATUS ROUTINES 1/15/87)
1
2 HEX
3 CODE clear.status ( m -- use mask to clear status bits)
4     1 POP          ( bit test mask)
5     SENSORSTAT # W MOV   ( load address )
6     2 W) 1 TEST O=    ( if acknowledge bit is clear )
7         IF FFFF # 1 XOR  ( form mask)
8         1 W ) AND      ( clear flag in status word)
9         THEN NEXT
10
11 .S
12
13
14
15

```

433 LIST

```

0 ( McCLELLAN BUILT IN TEST STATUS ROUTINES 1/15/87)
1
2 HEX
3 CODE set.status ( m -- use mask m to set bit in status)
4     1 POP          ( bit test mask)
5     SENSORSTAT # W MOV   ( load address)
6     W ) 1 TEST O=    ( is status bit clear ?)
7         IF 1 W ) OR    ( set status bit )
8         1 2 W) OR      ( set acknowledge bit )
9         THEN NEXT
10 DECIMAL
11 : NMI.OFF? ( -- f RETURN TRUE FLAG IF COUNT IS OVER LIMIT)
12     0 NMICHKCNT @ 2500 >      ( COUNT OVERLIMIT ? )
13     IF NOT           ( YES, RETURN TRUE )
14     ELSE 1 NMICHKCNT +!      ( INCREMENT COUNT )
15     THEN ;       .S

```

434 LIST

```

0 ( SENSOR SYSTEM SELF-TEST ROUTINE      2/FEB/87 )
1
2 HEX
3 CODE micro.test ( -- n, TESTS MICROPROCESSOR OPERATIONS )
4     5555 # 0 MOV 0 1 MOV      ( CX = AX = 5555 )
5     AAAA # 2 MOV 2 0 ADD  O<  ( AX = FFFF ? )
6     IF 1 0 XOR 0 SHR      ( AX = 5555 )
7     THEN 0 1 SUB  O=      ( CX = 0 ? )
8     IF 2 SHL 2 0 SUB      ( CX = 1 )
9     THEN 1 0 OR 0 PUSH      ( CX = 1 )
10    NEXT
11
12
13 .S
14
15

```

435 LIST

```
0 ( SENSOR SYSTEM SELF-TEST ROUTINE          2/FEB/87 )
1
2 CODE hit.dog ( PULSES WATCHDOG TIMER TO AVOID MICRO RESET )
3     LASTDAC 0 MOV B           ( use last dac value)
4     watchdog OUT NEXT
5
6 : @.VERROR ( -- n, RETURNS n TIMES VOLT TEST FAILED )
7     VERROR DUP C@ 1+ 2 MIN DUP ROT C! ;
8
9 : TEST.uP ( PERFORMS SYSTEM 8088 MICROPROCESSOR TEST )
10    1 micro.test 1 =           ( uPROC. OKAY ? )
11    IF clear.status          ( FLAG TEST PASSED )
12    ELSE set.status          ( FLAG TEST FAILED )
13    THEN hit.dog PAUSE ;
14 .S
15
```

436 LIST

```
0 ( SENSOR SYSTEM SELF-TEST ROUTINE          2/FEB/87 )
1
2 HEX
3 CODE ram.test ( a -- f, TEST SYSTEM RAM FOR WRITE & READ ERRORS)
4     0 # 2 MOV 8 # 1 MOV W POP      ( CK 8 BYTES BLOCK )
5     BEGIN CLI W ) 0 MOV B       ( SAVE RAM BYTE CONTENT)
6         55 #B W ) MOV          ( WRITE '55' PATTERN )
7         55 #B W ) CMP 0=        ( '55' PATTERN OK ? )
8         IF AA #B W ) MOV          ( WRITE 'AA' PATTERN )
9         AA #B W ) CMP 0= NOT    ( 'AA' PATTERN BAD ? )
10        IF 2 INC THEN          ( FLAG RAM ERROR )
11        ELSE 2 INC             ( FLAG RAM ERROR )
12        THEN 0 W ) MOV B       ( RESTORE RAM CONTENT )
13        W INC STI             ( POINT TO NEXT BYTE )
14        LOOP 2 PUSH NEXT
15 .S
```

437 LIST

```
0 ( SENSOR SYSTEM SELF-TEST ROUTINE          2/FEB/87 )
1 HEX
2 : TEST.RAM ( PERFORMS SYSTEM 8K RAM WRITE/READ TEST )
3     0 RAMFLAG C!           ( CLEAR TEST FLAG )
4     4 0 B00 0              ( SETUP POINTERS )
5     DO BEGIN PAUSE hit.dog
6         RAMFLAG C@          ( START TEST ? )
7         NMI.OFF? OR          ( NO NMI's ? )
8         UNTIL 0 RAMFLAG C!   ( CLEAR TEST FLAG )
9         I ram.test DUP      ( ANY BYTE FAILED ? )
10        IF LEAVE            ( STOP TEST )
11        THEN + hit.dog PAUSE 8 ( SET TO NEXT 8 BYTES )
12        +LOOP                ( RESULT ON STACK TOP )
13        IF set.status        ( FLAG TEST FAILED )
14        ELSE clear.status    ( FLAG TEST PASSED )
15        THEN hit.dog PAUSE ; .S
```

438 LIST

```
0 ( SENSOR SYSTEM SELF-TEST ROUTINE      2/FEB/87 )
1
2 HEX
3 CODE rom.test ( n a -- n, RETURNS SUMMATION OF 8 BYTES
4                                LOCATED AT EPROM ADDRESS a )
5           I W MOV I POP 2 POP          ( I POINTS TO a )
6           0 0 SUB 8 # 1 MOV          ( SETUP FOR 8 BYTES)
7           BEGIN LODS B 0 2 ADD      ( GET BYTE AND ADD )
8           LOOP W I MOV 2 PUSH NEXT ( RETURN RESULT )
9
10 CODE check.power ( -- f USED TO CHECK POWER SUPPLY FLAG, PFAIL*)
11           0 0 SUB ad/csr 0 MOV B    ( CLEAR 0, READ REGISTER)
12           80 #B 0 AND             ( SAVE ONLY MSB)
13           0 PUSH NEXT            ( RETURN RESULTS)
14 .S
15
```

439 LIST

```
0 ( SENSOR SYSTEM SELF-TEST ROUTINE      2/FEB/87 )
1
2 HEX
3 : TEST.ROM ( PERFORMS SYSTEM 16K EPROM CHECKSUM TEST )
4           2 0 FFFF8 C000          ( SETUP POINTERS )
5           DO I rom.test          ( ADD 8 BYTES BLOCK )
6           hit.dog PAUSE 8        ( SETUP FOR NEXT BLOCK)
7           +LOOP FFFE @ =         ( SUM = EPROM CKSUM ? )
8           IF clear.status       ( FLAG TEST PASSED )
9           no.op1                ( *** REMOVE LATER *** )
10          ELSE set.status        ( FLAG TEST FAILED )
11          THEN hit.dog PAUSE ;
12
13
14 .S
15
```

440 LIST

```
0 ( SENSOR SYSTEM SELF-TEST ROUTINE      3/24/87 )
1
2 CODE read.timer ( -- n, n read 8254 to see if its operating)
3           CLI                  ( disable interrupts )
4           00 #B ckcsr MOV       ( latch ce0 )
5           ce0 1 MOV B          ( read lsb )
6           ce0 0 MOV B          ( read msb )
7           0 1 HI MOV B 1 PUSH   ( combine )
8           4B # 1 MOV           ( setup loop )
9           BEGIN LOOP          ( wait )
10          00 #B ckcsr MOV       ( latch ce0 )
11          ce0 1 MOV B          ( read lsb )
12          ce0 0 MOV B          ( read msb )
13          0 1 HI MOV B          ( combine )
14          STI 1 PUSH NEXT      ( enable interrupts )
15 .S
```

441 LIST

```
0 ( SENSOR SYSTEM SELF-TEST ROUTINE      3/24/87 )
1
2 : TEST.COUNTER ( PERFORM TEST ON COUNTER )
3   0 RAMFLAG C!
4     BEGIN PAUSE hit.dog                ( CLEAR SYNC FLAG )
5       RAMFLAG C@                      ( SYNC WITH NMI ? )
6       NMI.OFF? OR                     ( NO NMI's ? )
7       UNTIL 32 read.timer             ( READ TIMER TWICE )
8         2DUP UK NOT                 ( CTR HASN'T CROSSED 0 ? )
9         IF - 2 6 WITHIN             ( WITHIN RANGE ? )
10        NMI.OFF? NOT AND           ( NO NMI's ? )
11        IF clear.status            ( CLEAR CTR ERROR )
12        ELSE set.status            ( FLAG COUNTER ERROR )
13        THEN
14        ELSE 2DROP DROP             ( DROP TIMER #s & MASK )
15        THEN hit.dog PAUSE ; .S
```

442 LIST

```
0 ( McCLELLAN - SENSOR POWER SUPPLY RANGE TEST 7/28/86)
1
2 : TEST.SUPPLIES ( -- TEST VOLTAGE RANGE FOR % TOLERANCE )
3   16384                            ( ERROR BIT MASK )
4   check.power                      ( READ STATUS FLAG )
5   IF clear.status                 ( CLEAR PWR SUPPLY ERR )
6     0 VERROR C!
7   ELSE @.VERROR 2 =               ( CLEAR ERROR COUNT )
8     IF set.status                  ( CHECK ERROR COUNT )
9     ELSE DROP                      ( FLAG PWR SUPPLY ERR )
10    THEN
11    THEN hit.dog PAUSE ; .S
12
13
14
15
```

443 LIST

```
0 ( SENSOR SYSTEM SELF-TEST ROUTINE      2/FEB/87 )
1
2 : TEST.SYSTEM ( TESTS SYSTEM EPROM, RAM, AND MICROPROCESSOR )
3   BEGIN hit.dog PAUSE              ( PULSE WATCHDOG )
4     TEST.ROM                      ( CHECK 16K EPROM CKSUM )
5     TEST.COUNTER                  ( CHECK 8224 TIMER )
6     TEST.SUPPLIES                 ( CHECK POWER SUPPLIES )
7     TEST.RAM                      ( CHECK 8K RAM )
8     TEST.uP                       ( CHECK 8088 MICROPROCESR )
9     AGAIN ;
10
11
12
13
14
15
```

450 LIST

```
0 ( OPERATOR VECTORED TERMINAL ROUTINES           2/MAY/85 )
1
2 HEX
3 CODE clr.uart ( CLEARS USR & RBR OF MICROBD'S UART )
4           comm/csr 0 MOV B  comm/tr 0 MOV B  NEXT
5
6 CODE mcr! ( n -- PROGRAMS UART MCR WITH n )
7           0 POP  0 comm/mcr MOV B  NEXT
8
9 CODE int#9 ( n -- ASSEMBLES INTERRUPT n )
10          9 INT  NEXT
11
12
13 : (EXPECT)  clr.uart 25 mcr! STOP ;
14
15
```

451 LIST

```
0 ( TEST CODE ROUTINES)      HEX
1
2 : (PAGE)    OC ( FF) EMIT ;
3 : (CR)     OD EMIT ;
4 : LF     OD EMIT  OA EMIT ;
5
6 : (TYPE)   clr.uart  25 mcr!          ( SETUP TO XMIT)
7           PTR @ C@  comm/tr C!          ( OUTPUT FIRST CHARACTER)
8           1 PTR +!  CTR @ 1- CTR ! STOP ; ( UPDATE PTR & CTR)
9
10
11
12
13
14
15
```

452 LIST

```
0 ( SENSOR INTER-ASSY COMMUNICATION TASK ROUTINE )
1
2 : (TYPE) ( INTER-ASSY VECTOR TYPE ROUTINE )
3           PTR @ C@ 1 PTR +!          ( GET FIRST CHARACTER )
4           CTR @ 1- CTR !          ( DECREMENT CTR )
5           comm/tr C! STOP ;       ( O/P CHAR )
6
7 : (EXPECT) ( INTER-ASSY VECTOR EXPECT ROUTINE )
8           STOP ;
9
10
11 .S
12
13
14
15
```

453 LIST

```
0 ( SENSOR INTER-ASSY COMMUNICATION TASK ROUTINE )
1
2
3
4 .S
5
6
7
8
9
10
11
12
13
14
15
```

454 LIST

```
0 ( SENSOR INTER-ASSY COMMUNICATION TASK ROUTINE )
1
2 CODE mcr! ( n -- PROGRAMS 8252 MCR REGISTER WITH n )
3     0 POP  0 comm/mcr MOV B  NEXT
4
5 CODE carr? ( -- f, RETURNS STATUS OF CARRIER ON DSR INPUT )
6     comm/msr 0 MOV B  2 # 0 AND  ( GET DSR STATUS )
7     2 #B 0 XOR  0 PUSH  NEXT    ( INVERT, TRUE = CARRIER)
8
9
10
11 .S
12
13
14
15
```

455 LIST

```
0 ( SENSOR INTER-ASSY COMMUNICATION TASK ROUTINE )
1
2 HEX
3 : CARR.ON ( TURNS ON MODEM CARRIER IF NO LINE ACTIVITY )
4     BEGIN PAUSE carr? 0=          ( NO CD ? )
5     UNTIL 25 mcr! 3 0           ( TURN CARRIER ON )
6     DO FF PAD I + C!
7     LOOP PAD 3 TYPE ;
8
9 : CARR.OFF ( TURNS MODEM CARRIER OFF )
10    24 mcr!                   ( TURN CARRIER OFF )
11    BEGIN PAUSE carr? 0=          ( NO CD ? )
12    UNTIL 64 mcr! ;
13
14 .S
15
```

456 LIST

```
0 ( SENSOR INTER-ASSY COMMUNICATION TASK ROUTINE )
1
2 : TEST.XMIT ( TESTS XMITTER BY LOOPBACK CHAR )
3     64 CHARBUFF C@           ( GET LOOPBACK CHAR )
4     16 =                   ( IS IT 'DLE' CHAR ? )
5     IF clear.status        ( FLAG NO XMIT ERR )
6         O UARTErr C!
7     ELSE UARTErr C@ 1+      ( CLEAR ERROR CTR )
8         3 MIN DUP          ( INC. ERROR CTR )
9         UARTErr C! 3 =      ( NO MORE THAN 3 )
10        IF set.status       ( FLAG XMIT ERROR )
11        ELSE clear.status   ( FLAG NO XMIT ERR )
12        THEN
13        THEN O CHARBUFF C! ;
14
15 .S
```

457 LIST

```
0 ( SENSOR INTER-ASSY COMMUNICATION TASK ROUTINE )
1
2 HEX
3 CODE crc-16 ( n a -- bcc, RETURNS CRC-16 BCC FOR n CHARS AT a )
4     W POP 0 # 0 MOV 0 # 2 MOV
5     BEGIN W ) 2 MOV B 2 PUSH B # 1 MOV
6     BEGIN 2 POP 2 PUSH 0 2 XOR 1 # 2 AND 0=
7     IF 0 SHR
8     ELSE 2 SHR 0 RCR 2001 # 0 XOR
9     THEN 2 POP 2 SHR B 2 PUSH
10    LOOP 2 POP W INC 2 POP 2 DEC 2 PUSH 0=
11    UNTIL 2 POP 0 PUSH NEXT      ( RETURN CRC-16 BCC )
12
13
14 .S
15
```

458 LIST

```
0 ( SENSOR INTER-ASSY COMMUNICATION TASK ROUTINE )
1
2 : COMM.XMIT ( a -- TRANSMITS POLL/REPOLL RESPONSE )
3     CARR.ON           ( TURN CARRIER ON )
4     COMMBUFF 1+ -      ( # OF CHARS TO SEND )
5     DUP 2- CHARCNT C! ( SAVE #-2 FOR LOOPBACK TEST )
6     3 O
7     DO COMMBUFF 1+
8     OVER TYPE          ( SEND RESPONSE )
9     LOOP DROP TEST.XMIT ( TEST XMIT LOOPBACK )
10    CARR.OFF ;         ( TURN CARRIER OFF )
11
12 : !CHAR ( a c -- a+1, STORES CHAR c AT a )
13     OVER C! 1+ ;
14 : !VAR ( a v -- a+2, STORES VARIABLE v AT a )
15     OVER ! 2+ ; .S
```

459 LIST

```
0 ( SENSOR INTER-ASSY COMMUNICATION TASK ROUTINE )
1
2 : SEND.REPORT ( SEND REPORT OF ENTRNAL VALUES & S/N#'s)
3     CARR.ON TABLES 7 TYPE ( SEND S/N #'s)
4     COMMBUFF 1+ 1 !CHAR
5     2 !CHAR 3 !CHAR           ( FORM 1, 2, 3 LEADER)
6     ELEMENTS 12 SET          ( MOVE DATA TO COMMBUFF)
7     DO I @ !VAR 2
8     +LOOP PRESSURE @ !VAR SPURCNT @ !VAR
9     SENSORSTAT @ !VAR NMICHKCNT @ !VAR
10    BTEMPBAR @ 2 n/ !VAR ELEMENTPOWER 8 SET ( MOVE MORE )
11    DO I @ !VAR 2
12    +LOOP 13 !CHAR 10 !CHAR ( FAKE CRC-16 = CR,LF)
13    COMMBUFF 1+ - COMMBUFF 1+ ( COMPUTE CHARACTER COUNT )
14    SWAP TYPE CARR.OFF ; .S ( SEND MESSAGE)
15
```

460 LIST

```
0 ( SENSOR INTER-ASSY COMMUNICATION TASK ROUTINE )
1
2 : STORE.DATA ( a -- a, STORES X,Y, & STATUS INTO COMMBUFF )
3     3 0
4     DO POLLTEMP I 2* + @      ( GET DATA FROM TEMP BUFF)
5     256 /MOD SWAP           ( SPLIT INTO MSB & LSB )
6     ROT 2 0
7     DO OVER !CHAR           ( STORE DATA CHAR )
8     SWAP 16 =
9     IF 16 !CHAR              ( DATA = DLE CHAR ? )
10    THEN
11    LOOP PAUSE
12    LOOP ;
13 .S
14
15
```

461 LIST

```
0 ( SENSOR INTER-ASSY COMMUNICATION TASK ROUTINE )
1
2 : RESP.XMIT ( n -- TRANSMITS RESPONSE WITH OPCODE n )
3     COMMBUFF 1+ 16 !CHAR       ( STORE DLE CHAR )
4     2 !CHAR                  ( STORE STX CHAR )
5     SWAP !CHAR               ( STORE OPCODE )
6     POLLCNT C@ !CHAR         ( STORE POLL COUNT )
7     STORE.DATA               ( STORE X,Y, & STATUS )
8     16 !CHAR 3 !CHAR          ( STORE DLE & ETX CHARS )
9     DUP COMMBUFF 3 + -
10    COMMBUFF 3 + crc-16        ( # OF CHARS FOR CRC-16 )
11    OVER ! 2+
12    255 !CHAR                ( STORE CALCULATED CKSUM )
13    COMM.XMIT ;              ( STORE PAD CHAR )
14
15 .S
```

462 LIST

```
0 ( SENSOR INTER-ASSY COMMUNICATION TASK ROUTINE )
1
2 : POLL.RES ( a o -- o, GETS NEW X,Y, & STATUS VALUE FOR RESP )
3     2+                               ( FORMAT OPCODE )
4     SWAP 1+ C@ POLLCNT C!           ( GET & SAVE POLL # )
5     XBAR @ POLLTEMP !             ( GET & SAVE X VALUE )
6     YBAR @ POLLTEMP 2+ !          ( GET & SAVE Y VALUE )
7     SENSORSTAT @                 ( GET STATUS )
8     POLLTEMP 4 + !               ( SAVE STATUS )
9     O SENSORSTAT 2+ ! ;         ( ACKNOWLEDGE STATUS )
10
11 .S
12
13
14
15
```

463 LIST

```
0 ( SENSOR INTER-ASSY COMMUNICATION TASK ROUTINE )
1
2 : REPOLL.RES ( a o -- o, FORMATS REPOLL RESPONSE )
3     2+                               ( FORMAT OPCODE )
4     SWAP 1+ C@                   ( GET REQ. POLL # )
5     POLLCNT C@ = NOT            ( NOT SAME AS LAST # ? )
6     IF 2- STXPTR @ 1+
7         SWAP POLL.RES           ( GET NEW DATA FOR RESP )
8     THEN ;
9
10
11
12
13
14 .S
15
```

464 LIST

```
0 ( SENSOR INTER-ASSY COMMUNICATION TASK ROUTINE )
1
2 : RESPOND ( FORMATS & SENDS POLL/REPOLL RESPONSE )
3     STXPTR @ 1+ DUP C@ DUP      ( GET OPCODE )
4     3 AND 2 =                  ( REPOLL REQUEST ? )
5     IF REPOLL.RES              ( FORMAT REPOLL RESPONSE )
6     ELSE POLL.RES              ( FORMAT POLL RESPONSE )
7     THEN RESP.XMIT ;          ( TRANSMIT RESPONSE )
8
9 : SEND.S/N ( SEND SERIAL NUMBERS )
10    CARR.ON TABLES 64 TYPE     ( TYPE NUMBERS )
11    CARR.OFF ;
12 .S
13
14
15
```

465 LIST

```
0 ( SENSOR INTER-ASSY COMMUNICATION TASK ROUTINE )
1 HEX
2 : POLL.REQ? ( -- f, CHECKS IF POLL/REPOLL REQUEST )
3     STXPTR @ 1+ C@ DUP           ( GET OPCODE )
4     30 AND 10 / CONFIG C@ 3 AND = ( SAME SENSOR # ? )
5     IF DUP 7 AND DUP 1 =        ( POLL OR )
6         SWAP 2 = OR NOT          ( REPOLL ? )
7         IF OF AND DUP 7 =        ( REPORT COMMAND ? )
8             IF DROP SEND.REPORT ( YES, SEND REPORT )
9             ELSE 6 =              ( SERIAL # COMMD ? )
10            IF SEND.S/N          ( YES, SEND #'S )
11            THEN
12            THEN 0                ( FLAG NOT POLL/REPOLL )
13            THEN
14            ELSE NOT              ( NO MATCH, RETURN FALSE )
15            THEN ; .S
```

466 LIST

```
0 ( SENSOR INTER-ASSY COMMUNICATION TASK ROUTINE )
1
2 : CRC-16? ( -- f, DETER. IF CRC-16 BCC SENT IS VALID )
3     ETXPTR @ STXPTR @ 2DUP -      ( SETUP COUNT & PTR )
4     SWAP 1+ crc-16 PAUSE          ( CALCULATE CRC-16 )
5     SWAP 1+ @                      ( GET CRC-16 IN BUFFER )
6     = ;
7
8
9
10 .S
11
12
13
14
15
```

467 LIST

```
0 ( SENSOR INTER-ASSY COMMUNICATION TASK ROUTINE )
1
2 : FIND.ETX ( a -- f, FINDS POSITION OF "ETX" CHAR IN COMMBUFF )
3     PAUSE 0 ETXPTR !
4     COMMBUFF 16 + SWAP           ( START AT a )
5     DO I C@ 16 =
6         IF I 1+ C@ 3 =          ( DLE CHAR ? )
7             IF I 1+ ETXPTR !    ( ETX CHAR ? )
8                 1 LEAVE          ( POINT TO ETX CHAR )
9                 ELSE 2           ( FOUND ETX CHAR )
10                THEN
11                ELSE 1           ( SKIP NEXT CHAR )
12                THEN
13                +LOOP ETXPTR @ PAUSE ; ( TRY NEXT CHAR )
14
15 .S
```

468 LIST

```
0 ( SENSOR INTER-ASSY COMMUNICATION TASK ROUTINE )
1
2 : FIND.STX ( a -- f, FINDS POSITION OF "STX" CHAR IN COMMBUFF )
3     PAUSE 0 STXPTR !
4     COMMBUFF 8 + SWAP           ( START AT a )
5     DO I C@ 16 =
6         IF I 1+ C@ 2 =          ( DLE CHAR ? )
7             IF I 1+ STXPTR !    ( STX CHAR ? )
8                 1 LEAVE        ( FOUND STX CHAR )
9                 ELSE 2          ( SKIP NEXT CHAR )
10                THEN
11                ELSE 1          ( TRY NEXT CHAR )
12                THEN
13            +LOOP STXPTR @ PAUSE ; ( RETURN STX POSITION )
14
15 .S
```

469 LIST

```
0 ( SENSOR INTER-ASSY COMMUNICATION TASK ROUTINE )
1
2 : FIND.MESS ( -- f, LOCATES MESSAGE & CHECKS CRC-16 CKSUM )
3     COMMBUFF 1+ FIND.STX DUP   ( FIND DLE/STX CHARS ? )
4     IF 1+ FIND.ETX            ( FIND DLE/ETX CHARS ? )
5     IF CRC-16?
6         ELSE 0                ( CK IF CRC-16 VALID )
7         THEN
8     THEN ;
9
10
11 .S
12
13
14
15
```

470 LIST

```
0 ( SENSOR INTER-ASSY COMMUNICATION TASK ROUTINE )
1
2 : PROCESS.BUFF ( VERIFIES CRC-16 & IF POLL/REPOLL REQUEST )
3     FIND.MESS              ( LEGAL REQUEST ? )
4     IF POLL.REQ?            ( POLL OR REPOLL REQ. ? )
5     IF RESPOND              ( RESPOND TO REQUEST )
6     THEN
7     THEN ;
8
9
10
11 .S
12
13
14
15
```

471 LIST

```
0 ( SENSOR INTER-ASSY COMMUNICATION TASK ROUTINE )
1
2 : INTER.ASSY ( HANDLES COMMUNICATION WITH INDICATOR/RECORDER )
3     BEGIN COMMBUFF 40 ERASE          ( CLEAR REC. BUFFER )
4         COMMBUFF 1+ 39 EXPECT        ( SETUP FOR RECP. )
5         COMMBUFF C@                ( REC. REQUEST ? )
6         IF PROCESS.BUFF           ( PROCESS REQ. )
7         THEN PAUSE
8         AGAIN ;
9
10 .S
11
12
13
14
15
```

472 LIST

```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
```

473 LIST

```
0 ( SENSOR INTER-ASSY COMMUNICATION TASK ROUTINE )
1 HEX
2 : POLL.REQ? ( -- f, CHECKS IF POLL/REPOLL REQUEST )
3     STXPTR @ 1+ C@ DUP          ( GET OPCODE )
4     30 AND 10 / CONFIG C@ 3 AND = ( SAME SENSOR # ? )
5     IF DUP 7 AND DUP 1 =
6         SWAP 2 = OR NOT        ( POLL OR )
7         IF OF AND DUP 7 =
8             IF DROP SEND.REPORT ( YES, SEND REPORT )
9             ELSE 6 =
10            IF SEND.S/N        ( SERIAL # COMM'D ? )
11            THEN
12            THEN 0              ( FLAG NOT POLL/REPOLL )
13            THEN
14            ELSE NOT           ( NO MATCH, RETURN FALSE )
15            THEN ; .S
```

474 LIST

```

0 ( ( POLL.REQ? TEST )
1
2 CVARIABLE CONFIG
3 VARIABLE STXPTR
4 2VARIABLE STX+1
5
6 : SEND.REPORT 123 ;
7 : SEND.S/N 456 ;
8 : INTV 0 CONFIG C! STX+1 STXPTR ! ;
9 .S
10
11
12
13
14
15

```

475 LIST

```

0 ( SENSOR INTER-ASSY COMMUNICATION TASK ROUTINE )
1 HEX
2 : POLL.REQ? ( -- f, CHECKS IF POLL/REPOLL REQUEST )
3     STXPTR @ 1+ C@ DUP          ( GET OPCODE )
4     30 AND 10 / CONFIG C@ 3 AND =      ( SAME SENSOR # ?? )
5     IF DUP 7 AND DUP 1 = SWAP 2 = OR ( PL OR REPL REQ ? )
6     IF
7         ELSE OF AND DUP 7 =      ( REPORT COMMAND ? )
8             IF DROP SEND.REPORT ( YES, SEND REPORT )
9             ELSE 6 =           ( SERIAL # COMMAND )
10            IF SEND.S/N        ( YES, SEND #'S )
11            THEN
12            THEN 0           ( FLAG NOT POLL/REPOLL )
13            THEN
14            ELSE NOT
15            THEN ; .S

```

476 LIST

```

0 ( SENSOR INTER-ASSY COMMUNICATION TASK ROUTINE )
1 HEX
2 : POLL.REQ? ( -- f, CHECKS IF POLL/REPOLL REQUEST )
3     STXPTR @ 1+ C@ DUP          ( GET OPCODE )
4     7 AND DUP 1 = SWAP 2 = OR ( REPOLL REQ ? )
5     IF 30 AND 10 / CONFIG C@ 3 AND = ( SAME SENSOR # ?? )
6     ELSE 30 AND 10 / CONFIG C@ 3 AND =
7         IF OF AND DUP 7 =      ( REPORT COMMAND ? )
8             IF DROP SEND.REPORT ( YES, SEND REPORT )
9             ELSE 6 =           ( SERIAL # COMMAND )
10            IF SEND.S/N        ( SEND NUMBERS )
11            THEN
12            THEN
13            THEN 0           ( FLAG NOT POLL/REPOLL )
14            ELSE NOT
15            THEN ; .S

```

477 LIST

```
0 ( POLL. REQ? TEST PART 2 )
1
2 : POLL
3     INTV    33 1
4     DO I STX+1 1+ C!
5     POLL. REQ?   STXPTR @ 1+ C@ . . . S
6     LOOP .S ;
7
8
9
10
11
12
13
14
15
```

478 LIST

```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
```

479 LIST

```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
```

480 LIST

```
0 ( SENSOR INTERRUPT HANDLERS           1/14/87 )
1
2 HEX
3 LABEL ireturn ( INTERRUPT RETURN HANDLER )
4           U POP 1 POP 0 POP IRET
5
6
7 CODE DIVERR ( DIVIDE BY 0/OVERFLOW; INTERRUPT 0 VECTOR RTN )
8           0 # 0 MOV IRET
9
10
11 .S
12
13
14
15
```

481 LIST

```
0 ( SENSOR INTERRUPT HANDLERS           1/14/87 )
1
2 HEX
3 LABEL <INTRO> ( SPURIOUS INTERRUPT HANDLER )
4           SPURCNT INC IRET
5
6
7
8
9
10
11
12
13 .S
14
15
```

482 LIST

```
0 ( SENSOR INTERRUPT HANDLERS           1/14/87 )
1
2 HEX
3 LABEL <INTR1> ( 8252 UART 'DATA READY' INTERRUPT HANDLER )
4           0 PUSH 1 PUSH U PUSH          ( SAVE REGS )
5           'OPERATOR # U MOV comm/tr 0 MOV B          ( GET CHAR )
6           80 #B CTR U) TEST 0<          ( EXPECTING CHARS ? )
7           IF PTR U) W XCHG 0 W ) MOV B          ( STORE CHAR )
8           W INC PTR U) W XCHG          ( INC. PTR )
9           CNT U) INC CTR U) INC 0=          ( CTR = 0 ? )
10          IF 0 #B COMMBUFF MOV          ( FLAG BAD BUFFER )
11          WAKE # STATUS U) MOV          ( AWAKEN I/A TASK )
12          THEN
13
14 .S
15
```

483 LIST

```
0 ( SENSOR INTERRUPT HANDLERS           1/14/87 )
1
2 HEX
3     ELSE 1 #B TESTFLAG TEST  0= NOT      ( TESTING LOOPBACK ? )
4         IF 0 CHARBUFF MOV B               ( SAVE LOOPBACK CHAR )
5             0 #B TESTFLAG MOV
6             THEN
7                 THEN ireturn JMP
8
9
10
11
12
13
14 .S
15
```

484 LIST

```
0 ( SENSOR INTERRUPT HANDLERS           1/14/87 )
1
2 HEX
3 LABEL <xmit> ( 8252 UART 'TRANSMIT COMPLETE' INTERRUPT HANDLER )
4     CTR U) 0 MOV 0 0 OR 0                ( HAVE CHARS TO XMIT ?)
5     IF PTR U) I XCHG LODS B              ( GET CHAR FROM BUFFER)
6         PTR U) I XCHG 0 comm/tr MOV B   ( XMIT CHAR )
7         CTR U) DEC 0=                   ( XMIT LAST CHAR ? )
8         IF WAKE # STATUS U) MOV
9             ELSE CTR U) 0 MOV B          ( AWAKEN I/A TASK )
10            0 CHARCNT CMP B 0=          ( GET # OF CHARS )
11            IF 1 #B TESTFLAG MOV       ( 1ST CHAR XMITTED ? )
12            THEN
13            THEN
14            THEN RET
15 .S
```

485 LIST

```
0 ( SENSOR INTERRUPT HANDLERS           1/14/87 )
1
2 HEX
3 LABEL <INTR2> ( 8252 UART 'INTR' INTERRUPT HANDLER )
4     0 PUSH 1 PUSH U PUSH                ( SAVE REGS )
5     'OPERATOR #' U MOV comm/csr 0 MOV B ( GET STATUS )
6     0 1 MOV B 20 #B 0 TEST 0= NOT      ( XMIT COMPLETE ? )
7     IF <xmit> CALL
8     THEN 10 #B 1 TEST 0= NOT           ( MODEM STATUS ? )
9     IF comm/msr 0 MOV B
10        2 #B 0 AND                  ( MASK & INVERT DSR )
11        2 #B 0 XOR 0=                ( CD NOT PRESENT ? )
12        IF 2 #B CDFLAG TEST 0= NOT    ( CARR PRESENT BEFORE? )
13
14 .S
15
```

486 LIST

```
0 ( SENSOR 8252 "INTR" INTERRUPT HANDLER           7/28/86 )
1
2 HEX
3     IF 1 #B COMMBUFF MOV          ( FLAG RECD. XMISSION )
4         0 # CTR U) MOV          ( CLR REC. BUFFER CTR )
5         WAKE # STATUS U) MOV          ( AWAKEN I/A TASK )
6     THEN
7     THEN 0 CDFLAG MOV B          ( SET/RESET CD FLAG )
8     THEN ireturn JMP
9
10
11
12
13 .S
14
15
```

487 LIST

```
0 ( SENSOR NMI TIMER INTERRUPT HANDLER           7/28/86 )
1
2 HEX
3 LABEL <timer>  ( 8254 timer service routine)
4     0 PUSH    1 PUSH    U PUSH    PUSHF
5     PHASE 0 MOV B  0 INC
6     07 # 0 AND  0 PHASE MOV B      ( mod B ctr)
7     1 #B ACQUIRE MOV          ( START DATA ACQUISITION )
8     1 #B RAMFLAG MOV          ( SYNC START OF RAM TEST )
9     0 # NMICHKCNT MOV          ( SET NMI CHECK FLAG )
10    POPF                  ( RESTORE FLAGS)
11    ireturn JMP
12
13
14 .S
15
```

488 LIST

498 LIST

```
0  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15
```

499 LIST

```
0 ( McCLELLAN SENSOR  PRESSURE SNESOR POWER-UP ROUTINE  3/29/89)  
1  
2 : BARO.INIT ( INITIALIZE PRESSURE SAMPLE SUMS)  
3     B/T.ADJUST          ( A/D ADJUST SAMPLES)  
4     B M* BAROSUM 2!      ( SCALE UP AND SAVE)  
5     hit.dog  
6     B M* BTEMPSUM 2! ;   ( SCALE UP AND SAVE)  
7  
8  
9 .S  
10  
11  
12  
13  
14  
15
```

500 LIST

```
0 ( McCLELLAN SENSOR  ELEMENT POWER UP RAMP ROUTINES  12/1/86)  
1  
2 : GET TEMP  ( -- READ TEMPERATURE )  
3     A/DFAILCTR C@ 10 <           ( A/D FAILURES ?)  
4     IF O TEMPVOLT1 ! 0 VOLTS ! 4 O ( CLEAR VARIABLES)  
5     DO 4 a/d.sample TEMPVOLT1 +!  ( tempsense )  
6     6 a/d.sample VOLTS +!  
7     LOOP TEMPVOLT1 @ VOLTS @ 1STVOLTS 2! ( *****)  
8     CONVERT,V/T  
9     THEN ;  
10    HEX  
11 : SENSOR.ENABLE ( ENABLE THE WIND SENSORS )  
12     28 set.csrbbit ; ( SET THE ENABLE BITS )  
13  
14 .S  
15
```

501 LIST

```

0 ( McCLELLAN SENSOR ELEMENT POWER UP RAMP ROUTINES 12/1/86)
1
2 : END.POINTS ( t -- hp, lp DETERMINE END POINT OF RAMP )
3     0 INTERP.ELRES ( FIND STARTING POINT)
4     10000 +          ( SET OVERHEAT TO 100 DEG. C )
5     1 INTERP.ELRES DROP ( FIND ENDING POINT)
6     D/AOUTBUF 2@ ;
7
8 : TEST TEMP ( -- t TEST TEMPERATURE FOR VALID RANGE )
9     TEMPERATURE @ DUP -7000 < ( TEMP < -70 DEG C ?)
10    IF DROP 0           ( SET TO 0 DEG. C )
11    THEN ;
12 .S
13
14
15

```

502 LIST

```

0 ( McCLELLAN SENSOR ELEMENT POWER UP RAMP ROUTINES 9/22/87)
1
2 : 1ST.SETPT ( -- INITIALIAL SETTING OF ELEMENT TEMP )
3     A/DFAILCTR C@ 10 < ( A/D FAILURES ?)
4     IF 0 TEMPVOLT1 ! 0 VOLTS ! ( CLEAR VARIABLES)
5     0 VBARO ! 0 VBTEMP ! 4 0
6     DO A/D.READ           ( SAMPLE INPUT)
7     LOOP
8         VBARO @ VBTEMP @ 1STVOLTS 2! ( *****)
9         SET_ELEMENT_TEMP dac.load ( SET ELEMENT TEMP)
10        10000 0
11        DO hit.dog          ( ALLOW TO SETTLE )
12        LOOP
13        THEN ; .S
14
15

```

503 LIST

```

0 ( McCLELLAN SENSOR ELEMENT POWER UP RAMP ROUTINES 12/1/86)
1
2 : RAMP.GEN ( hp, lp -- OUTPUT A RAMP TO THE D/As )
3     DO I I 2DUP d/astuff
4     dac.load 120 0 ( OUTPUT VALUES )
5     DO hit.dog          ( HIT WATCHDOG WHILE WAITING)
6     LOOP
7     LOOP ;
8 .S
9
10
11
12
13
14
15

```

504 LIST

```
0 ( MALLELLAN SENSOR ELEMENT POWER UP RAMP ROUTINES 12/1/86)
1
2 : RAMP.UP ( -- READ TEMP & RAMP ELEMENTS TO OPERATING POINT)
3     GET TEMP TEST TEMP      ( READ TEMP & TEST IT)
4     hit.dog
5     END.POINTS           ( COMPUTE RAMP END POINTS )
6     DUP DUP 2DUP d/astuff
7     dac.load             ( LOAD A/D WITH 1p)
8     SENSOR.ENABLE RAMP.GEN ( ENABLE A/Ds & RAMP UP)
9     hit.dog
10    1ST.SETPT            ( SET ELEMENT TEMP )
11    BARO.INIT ;          ( INITIALIZE PRESSURE )
12
13
14 .S
15
```

505 LIST

```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
```

506 LIST

```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
```

507 LIST

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

508 LIST

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

509 LIST

0 (TASK USER VARIABLES INITIALIZATION 5/5/87)
1
2 HEX
3 I CREATE OPERATOR 'OPERATOR , EA2E , 'ACQUISITION , ZERO , 0 ,
4 'OPERATOR 81 - , 0A , ' (TYPE) , ' (EXPECT) ,
5
6 I CREATE ACQUISITION 'ACQUISITION , EA2E , 'PROCESS , ZERO ,
7 0 , 'ACQUISITION 41 - , 0A ,
8
9 I CREATE PROCESS 'PROCESS , EA2E , 'WDOG , ZERO , 0 ,
10 'PROCESS 41 - , 0A ,
11
12 I CREATE WDOG 'WDOG , EA2E , 'OPERATOR , ZERO , 0 ,
13 'WDOG 41 - , 0A ,
14
15 .S

525 LIST

```

0 ( SYSTEM POWER-UP ROUTINE                               30/APR/85 )
1
2 I : START [ RECOVER ]
3     hit.dog
4     INTERRUPTS [ ORAM 128 + ] LITERAL 16 MOVE
5     0 2 ! [ , DIVERR ] LITERAL 0 ! ( SET VECT 0 DIVERR)
6     0 10 ! [ (timer) ] LITERAL 8 ! ( SET NMI VECTOR)
7     ( SETUP RAM)
8     OPERATOR 2+ OPERATOR STATUS HIS DUP 60 ERASE 28 MOVE
9     ACQUISITION 2+ ACQUISITION STATUS HIS DUP 60 ERASE 12 MOVE
10    PROCESS 2+ PROCESS STATUS HIS DUP 60 ERASE 12 MOVE
11    WDOG 2+ WDOG STATUS HIS DUP 60 ERASE 12 MOVE
12    START-UP ;
13
14 .S
15

```

526 LIST

```

0 ( SYSTEM POWER-UP ROUTINE                               30/APR/85 )
1
2 HEX
3 LABEL POWER-UP 0 # 0 MOV
4             0 ES LSG   0 SS LSG   0 DS LSG
5             , START 2- # I MOV   OPERATOR U MOV   U R MOV
6             OPERATOR OA + S MOV   NEXT
7
8 HERE /PROM OF - ORG EA C, POWER-UP, ZERO,
9 ORG
10
11 .S
12
13
14
15

```

527 LIST

```

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

```